UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NET ASSESSMENT METHODOLOGIES AND CRITICAL DATA ELEMENTS FOR STRATEGIC AND THEATER FORCE COMPARISONS FOR TOTAL FORCE CAPABILITY ASSESSMENT (TFCA) (U): Volume III (continued)		5. TYPE OF REPORT & PERIOD COVERED Second Interim Report 6. PERFORMING ORG. REPORT NUMBER IDA PAPER P-1615, Vol. III
7. AUTHOR(s) Lowell Bruce Anderson Eleanor L. Schwartz		8. CONTRACT OR GRANT NUMBER(s) MDA 903 79 C 0018
9. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS T-0-071
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Chiefs of Staff (JCS) The Pentagon Washington, DC 20301		12. REPORT DATE January 1982 13. NUMBER OF PAGES 338
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DOD-IDA Management Office 400 Army-Navy Drive Arlington, Virginia 22202		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

N/A

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Combat model, naval combat, carrier task force, sea-based aircraft, land-based aircraft, bombers, escorts, surface ships, submarines, anti-air warfare, anti-submarine warfare, anti-surface ship warfare, airbase attack, power projection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The model described in this paper is an aggregated, fully automated, deterministic model of combat between two opposing forces. The Blue forces in this model can consist of aircraft carriers, escort ships, submarines, sea-based attack and defensive aircraft, and land-based defensive aircraft. The Red forces in the model can consist of surface ships, submarines, land-based attack and defensive aircraft, and ground defenses. The model is designed to simulate combat between these forces in areas in which geography can play a significant role, such as in the (continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4. TITLE (and Subtitle) continued

A Preliminary Documentation of a Naval Model - Second Interim Report

20. ABSTRACT (continued)

Mediterranean area. In particular, a goal in the design of this model was to include geographical considerations in an aggregated model--not to build a model that simulates either combat or geography (or both) in great detail. Thus, this model is intended to be appropriate for studies which can profitably use a comprehensive aggregated model, but which could not adequately use existing models either because these models did not address geography at all or because they were too detailed and insufficiently comprehensive. One purpose of this documentation is to describe the model sufficiently well that a prospective user can determine whether it would be a helpful tool to assist in analyzing any particular problem.

The description of the model in this paper is preliminary only in that portions of this description will be expanded in the near future in order to more thoroughly document the model. The current status of the model is as follows: Its programming is complete. That is, an input routine, the combat interaction routines, the output routines, and the code to hold these routines together have been programmed. An unclassified and entirely hypothetical data base has been prepared, the model has been successfully run with this data base, and brief initial tests of the model have been completed.

Copy 136 of 248 copies

IDA PAPER P-1615

NET ASSESSMENT METHODOLOGIES AND CRITICAL DATA ELEMENTS FOR STRATEGIC AND THEATER FORCE COMPARISONS FOR TOTAL FORCE CAPABILITY ASSESSMENT (TFCA) (U)

VOLUME III–A Preliminary Documentation of a Naval Model

Second Interim Report

Lowell Bruce Anderson
Eleanor L. Schwartz

January 1982

Prepared for
Joint Chiefs of Staff

INSTITUTE FOR DEFENSE ANALYSES
INTERNATIONAL SECURITY ASSESSMENT DIVISION

IDA Log No. HQ 81-24033

IDA PAPER P-1615

NET ASSESSMENT METHODOLOGIES AND CRITICAL DATA ELEMENTS FOR STRATEGIC AND THEATER FORCE COMPARISONS FOR TOTAL FORCE CAPABILITY ASSESSMENT (TFCA) (U)

VOLUME III–A Preliminary Documentation of a Naval Model

Second Interim Report

Lowell Bruce Anderson
Eleanor L. Schwartz

January 1982

IDA

INSTITUTE FOR DEFENSE ANALYSES
INTERNATIONAL SECURITY ASSESSMENT DIVISION
400 Army-Navy Drive, Arlington, Virginia 22202

Contract MDA 903 79 C 0018
Task 0-071

FOREWORD

The work reported in this paper is part of a sustained evaluation and study effort undertaken by the Institute for Defense Analyses to improve the analytic basis for worldwide Total Force Capability Assessments (TFCA)* conducted each year by the Joint Chiefs of Staff (Studies Analysis and Gaming Agency) in conjunction with the Services. A three-year program of work (begun in December 1979 under Task Order T-0-071), the TFCA studies were first reported on by an informal report of progress to JCS in March 1980, followed by the first interim report** of November 1980. Some of the topics presented in this report, in brief, were "wiring diagrams," an initial analytic expression of interconnections between many diverse elements of U.S. and NATO military forces as a potential basis for understanding dynamics of interactions of many diverse engagements; the use of a new "static measure" methodology, the Modified Antipotential Potential (MAP); and issues involving the Atlantic SEALOC campaign and protection of the SLOC; as well as modeling issues in the Central Front of Europe.

---

*The TFCA measures the capability of U.S. forces to carry out their missions under various assumed international circumstances and strategies.

**William J. Schultis, Lowell Bruce Anderson, Jerome Bracken, I. I. Deutsch, Rosemary Hayes Jones, Cynthia M. Hampton, H. Peter Liepman, Earl F. Pierson, Leonard Wainstein, *Net Assessment Methodologies and Critical Data Elements for Strategic and Theater Force Comparisons for Total Force Capability Assessment (TFCA): Interim Report (U)*, IDA Paper P-1536, November 1980, SECRET.

This paper is the third volume of the second interim report, IDA Paper P-1615,* consisting of four volumes which set forth the results of the second year of work. Other volumes are:

Volume I -- *Modifications for WVR/BVR Combat Modeling, Chemical Weapons Gaming, Chemical Weapons Statistic Measures, Concepts, and Cross-Employment.* This volume provides a summary and introduction for the entire second interim report and also contains sections on guidelines for gaming chemical weapons, data and static measures for assessment of the effects of the use of chemical weapons and cross employment of weapons.

Volume II -- *Illustrative Example of Static Measures and Methodology.* In this volume the Modified Antipotential Potential (MAP) method developed in the first interim report was extended to include improved methods of calculating the previous measures and to incorporate several new measures.

Volume IV -- *Preliminary Concept Formulation for Flow Model.* In this extension of the work reported in the first interim report using "wiring diagrams" and flow diagrams, a hypothetical global scenario was chosen to illustrate the type of analysis that might be supported by a flow model.

---

*Robert F. Robinson, Lowell Bruce Anderson, Stephen D. Biddle, Edward P. Kerlin, Rosemary Hayes Jones, Dale L. Moody, Leo A. Schmidt, William J. Schultis, Eleanor L. Schwartz, *Net Assessment Methodologies and Critical Data Elements for Strategic and Theater Force Comparisons for Total Force Capability Assessment (TFCA): Second Interim Report (U)*, Volume I (UNCLASSIFIED), Volume II (SECRET), Volumes III and IV (UNCLASSIFIED), IDA Paper P-1615, 1981-82.

CONTENTS

APPENDICES

FIGURES

TABLES

Chapter I

OVERVIEW

A. INTRODUCTION

The model described below is an aggregated, fully automated, deterministic model of combat between two opposing forces. Usually, these forces can be thought of as being NATO forces versus Warsaw Pact forces. However, since other applications are possible, the forces will be referred to as Blue forces and Red forces here. (In the NATO/Warsaw Pact context, NATO is Blue and the Warsaw Pact is Red.)

The Blue forces in this model can consist of aircraft carriers, escort ships, submarines, sea-based attack and defensive aircraft, and land-based defensive aircraft. The Red forces in the model consist of surface ships, submarines, land-based attack and defensive aircraft, and ground defenses. (A complete discussion of these resources is given below.) The model is designed to simulate combat between these forces in areas in which geography can play a significant role, such as in the Mediterranean area; and the first use of the model is expected to be an analysis of naval combat in the Mediterranean. The model described here was originally named MEDMOD (for Mediterranean Model) because of this intended initial use. However, the model is not inherently restricted to simulating combat in the Mediterranean, and the model can be used (with different inputs) to address naval combat in other areas. To emphasize that the model is not inherently restricted to the Mediterranean, the name of the model will

be changed to NAVMOD (for Naval Model). In particular, the model will be called MEDMOD in this preliminary documentation, and will be called NAVMOD in the final documentation to be written in the near future. (A few improvements will also be made; thus, there will be a few technical differences between the model described here, called MEDMOD, and the future version of the model, called NAVMOD.)

One goal in the design of MEDMOD is to include geographical considerations in an aggregated model--but not to build a model that simulates either combat or geography (or both) in great detail. Accordingly, the combat interactions and the geographical considerations are both relatively highly aggregated. Thus, MEDMOD is intended to be appropriate for studies which can profitably use a comprehensive aggregated model, but which could not adequately use existing models either because these models did not address geography at all or because they were too detailed and insufficiently comprehensive. One purpose of this documentation is to describe MEDMOD sufficiently well that a prospective user can determine whether MEDMOD would be a helpful tool to assist in analyzing any particular problem.

Much of MEDMOD was constructed by using concepts or computer code already developed for other models; this characteristic is discussed in greater detail in Section B, below. Sections C and D discuss the geography and resources, respectively, simulated in MEDMOD. Section E presents a brief overview of the major combat interactions simulated in MEDMOD. The current status of the MEDMOD computer program is discussed in Section F.

Chapter II presents the structure of the MEDMOD computer program and discusses each of the major interactions it

simulates in greater detail than Section E. Chapter III presents some limitations of MEDMOD.

B. BACKGROUND

The decision to build a comprehensive model of naval combat that includes geographic considerations was based, in part, on the fact that the Institute for Defense Analyses (IDA) had recently built an aggregated model of an air and submarine attack on a carrier task force. This model is described, documented, and used in IDA Report R-245 (Reference [1]), and will be called the R-245 model here. The R-245 model is similar in scope and aggregation to the model described in Reference [2] (and many of the data used in Reference [1] came from Reference [2]).

The computer program for the R-245 model was modified and converted into a subroutine of MEDMOD. This subroutine, called CTFMOD, is essentially the core of MEDMOD.

This documentation of MEDMOD does not require that the reader either has read R-245 or is familiar with the R-245 model. However, the reader who is interested in details concerning the CTFMOD portion of MEDMOD is urged to read the portions of R-245 which describe its model and data.

C. GEOGRAPHY

MEDMOD requires that the seas (and/or ocean) of interest be divided into a number of geographic regions (or, synonymously, locations). MEDMOD is currently dimensioned to hold up to six such regions--combat can take place in five of these regions with the remaining region being a "sanctuary" region. (The sanctuary region is labeled as Region 0.) Thus, for example, the Mediterranean area can be divided into the following six regions: (0) the Atlantic Ocean, (1) the Western Mediterranean (the portion of the Mediterranean west of

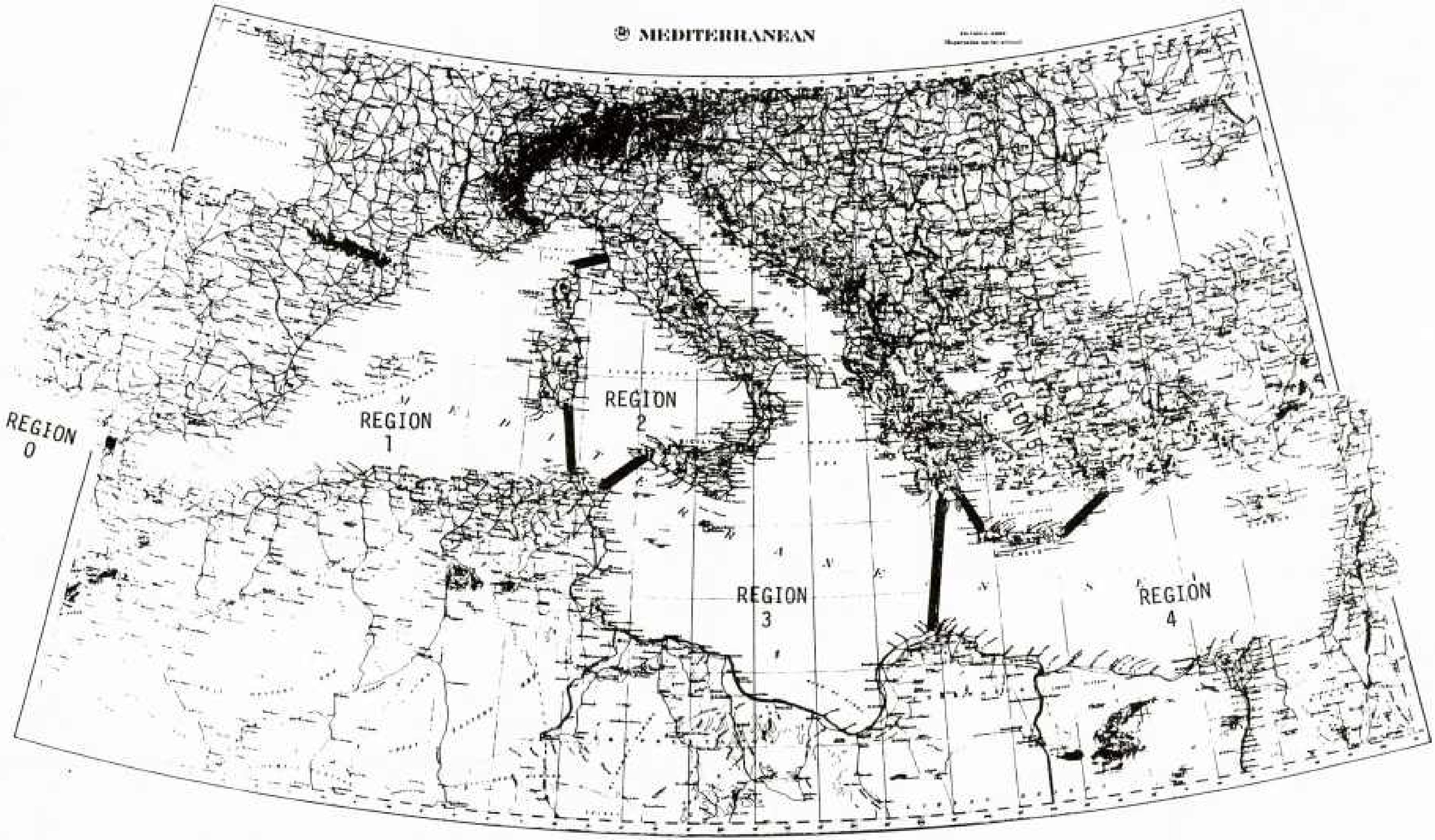


Figure 1. SAMPLE GEOGRAPHIC REGIONS FOR MEDMOD

Corsica and Sardinia), (2) the Tyrrhenian Sea, (3) the Central Mediterranean (the portion of the Mediterranean bounded by Italy, Tunisia, Libya, and Greece, including the Ionian Sea), (4) the Eastern Mediterranean (the part of the Mediterranean east of the shortest line connecting Libya and Greece), and (5) the Aegean Sea. This division of the Mediterranean is illustrated in Figure 1.

Relabeling these regions could allow the Adriatic Sea to be played instead of the Aegean or the Eastern Mediterranean, for example. Another relabeling could allow the Western Mediterranean to be split into two (or more) regions provided either that one (or more) of the other regions are deleted or that the dimensions of MEDMOD are expanded.

MEDMOD uses these regions in the following manner. First, the specific location of ships within any of these regions is not accounted for--MEDMOD only portrays that ships are somewhere inside these regions.

Second, Red surface ships can be in any region except Region 0, and those Red ships can simultaneously be in several or all of the regions (except Region 0). All Blue surface ships, however, are assumed to be in one Blue task force and, at any point in time, the task force is in one (and only one) of these regions, including Region 0.

Third, Red submarines can be in any (or all) regions (except Region 0), and can also be in barriers between regions. Blue submarines can be in direct support of the task force (i.e., in the same region as the task force), and can also be in barriers between regions.

Rules based on input parameters determine the movement of the Blue task force and any supporting submarines from region to region (these rules will be described later). For example, the task force might start in Region 0 and remain there for one time period, then move to Region 1 and remain

there for two time periods, then move to Region 2 for one time period, then move from Region 3 for four time periods, then move from Region 4 for three time periods, then move to Region 5 for the rest of the war (i.e., until an input maximum number of time periods has been simulated or until the task force loses all its effectiveness, as discussed below).

MEDMOD does not necessarily assume that the task force moves from west to east. Appropriate inputs would allow a task force to move back and forth between regions, or to start in Region 5 (for example) and attempt to escape to Region 0. However, for simplicity, MEDMOD assumes that a task force in Region i either remains in Region i, moves to Region i+1 (unless i is the largest numbered region being played), or moves to Region i-1 (unless i is 0). That is, the task force cannot move from Region i to Region j if $|i-j| \geq 2$ without spending at least one time period in each region between i and j. The reason for this restriction is to simplify the number of possible interactions between ships and submarines in barriers. If the Blue task force attempts to move from Region i to Region i+1 (or from Region i+1 to Region i), then the task force (and any submarines in direct support of it) will have to cross a Red submarine barrier if (and only if) there are Red submarines in a barrier between Region i and Region i+1. As currently dimensioned, MEDMOD plays up to six regions, so there can be up to five possible Red submarine barriers.

Similarly, another set of rules (described later) determines the movement of Red surface ships and non-barrier submarines from region to region. Red surface ships and non-barrier submarines can also move at most one region during a time period; and any Red ship or submarine attempting to move from Region i to Region i+1 (or from Region i+1 to Region i) will have to cross a Blue submarine barrier if and only if there are Blue submarines in a barrier between

Region i and Region i+1. As currently dimensioned, there can be up to five possible Blue submarine barriers. However, since Region 0 is a sanctuary, the movement rules in MEDMOD prohibit Red surface ships and submarines from entering Region 0, and so a Blue submarine barrier between Region 0 and Region 1 would be inactive.

Unlike surface ships or submarines in regions, Red and Blue submarines in barriers are never (automatically) moved to any other barrier or to inside any region by MEDMOD. (As described later, all resources can be moved by inputs, but Red and Blue barrier submarines can only be moved this way--they cannot automatically be moved by the MEDMOD code.)

Finally, many of the inputs for effectiveness parameters in MEDMOD are functions of the location of the task force. Thus, while the task force is in Region 2, for example, one set of effectiveness parameters is used; and if the task force moves to Region 3, a (potentially) different set of effectiveness parameters is automatically used.

D. RESOURCES

1. Blue Resources

Blue resources in MEDMOD can be grouped into four classes: surface ships, submarines, carrier-based aircraft, and land-based aircraft.

a. Surface Ships

MEDMOD simulates the following five types of Blue surface ships: aircraft carriers, antiair warfare (AAW) escort ships, air-capable (i.e., LAMPs-capable) antisubmarine warfare (ASW) escort ships, nonair-capable ASW escort ships, and underway replenishment ships. Multiple subtypes within these five types cannot be simulated in the same run of MEDMOD. However, one run of MEDMOD could be made that simulates one type of

aircraft carrier, for example, and another run of MEDMOD could be made (with different inputs) to simulate a different type of aircraft carrier--but two different types of carriers operating simultaneously cannot be simulated. There are no computer limitations on the number of ships in any of the five types. That is, virtually any number of ships in each of these five types can be input to MEDMOD. However, since MEDMOD simulates only one task force, all Blue surface ships must be part of the same task force.

MEDMOD does not require that these resources (or any Blue or Red resources) be strictly positive. That is, if a user of MEDMOD desires to not simulate any type of resource in a particular run, then this can be accomplished by inputting zero for the number of that type of resource in that run. For example, a task force consisting only of ASW destroyers protecting a region can be simulated by playing only these ships (so that the task force would have no carriers, AAW escorts, or replenishment ships).

b. Submarines

MEDMOD simulates one type of Blue submarine in direct support of the task force. These submarines move with the task force and so are always in the same region as the task force. MEDMOD simulates one type of Blue submarine in barriers between regions, and this type of submarine can be different than the Blue submarines in direct support of the task force. As stated above, Blue can have submarine barriers between any (or all) consecutively numbered regions. Again, there are no computer limitations on the number of Blue submarines of either type simulated in MEDMOD.

c. Carrier-Based Aircraft

MEDMOD plays the following four types of Blue carrier-based aircraft: attack aircraft, fighter aircraft, airborne early warning (AEW) aircraft, and ASW aircraft. Multiple subtypes within any of these four types of aircraft cannot be simulated in the same run of MEDMOD. There are no computer limitations on the number of aircraft in each of these four types. Of course, a user of MEDMOD should use inputs for the number of these types of aircraft that are consistent with the number (and type) of aircraft carriers being simulated. Since all carriers are in one task force, all carrier-based aircraft are always in the same region that the task force is in.

AEW aircraft provide warning (i.e., time and exact position) of the Red air raid so that fighters on the carriers assigned to deck launched intercept (DLI) missions can take off and be vectored to the attack, and fighters in the air on combat air patrol (CAP) missions can move toward the incoming raid. ASW aircraft form an outer ASW screen around the task force. Attack aircraft can be used on any or all of three missions: anti-surface ship warfare (to kill Red surface ships in the same region as the task force), airbase attack missions (to kill Red land-based aircraft), and power projection missions. MEDMOD simulates these first two interactions (Blue attack aircraft versus Red surface ships, and Blue attack aircraft on airbase attack missions) in a direct but aggregated manner. However, MEDMOD does not simulate ground-to-ground forces (such as tanks, infantry, or ground-to-ground artillery) on either side, thus the effects of Blue power projection missions are not simulated. Instead, the number of power projection missions successfully flown is an output measure of effectiveness of MEDMOD. This number of power projection missions successfully flown can be an absolute number or can be weighted by type of aircraft (Blue fighters

can also fly power projection missions) and by the geographic region that the task force was in when these missions were flown. Blue fighters can fly all missions that attack aircraft can fly, though (perhaps) at different levels of effectiveness. In addition, Blue fighters can defend the carrier by flying CAP or DLI missions as stated above, and can escort the aircraft that are attacking the notional vulnerable Red airbase.

d. Land-Based Aircraft

MEDMOD plays three classes of Blue land-based aircraft: AEW aircraft, ASW aircraft, and interceptors (strictly speaking, interceptors are fighter aircraft that are restricted to fly only intercept missions). There can only be one type of land-based AEW aircraft, but it can differ from the one type of carrier-based AEW aircraft in MEDMOD, and the same holds for ASW aircraft. MEDMOD can play multiple types of land-based interceptors--it is currently dimensioned to play up to two different types of these interceptors. Land-based Blue aircraft are associated with regions and a different number of these aircraft can be associated with each region. Thus, for example, there can be 10 land-based AEW aircraft, 20 land-based ASW aircraft, 30 interceptors of Type 1, and 40 interceptors of Type 2 associated with Region 1, and 15 land-based AEW, 25 land-based ASW aircraft, 35 interceptors of Type 1, and 40 interceptors of Type 2 associated with Region 2. Land-based AEW and ASW aircraft associated with a particular region contribute to the defense of the task force only when the task force is in that region. Land-based interceptors can help contribute to the defense of the task force no matter where the interceptors and task force are located; however, the degree of their contribution can depend on these locations. Again, there are no computer limitations on the numbers of land-based aircraft simulated in MEDMOD.

It should be noted that the R-245 model can simulate armed land-based AEW aircraft. This capability has been deleted in CTFMOD, and so MEDMOD can only simulate unarmed AEW aircraft. (Carrier-based AEW aircraft are always unarmed in the R-245 model and in MEDMOD.)

The missions of land-based AEW and ASW aircraft are the same as for the corresponding carrier-based aircraft. MEDMOD simulates only one mission for the Blue land-based interceptors, namely, to form an air barrier which Red bombers (and their escort aircraft, if any) may have to pass through in order to reach the Blue task force.

2. Red Resources

Red resources in MEDMOD can be grouped into four classes: surface ships, submarines, land-based aircraft, and ground defenses.

a. Surface Ships

MEDMOD does not simulate Red surface ships in the same degree of detail it simulates Blue surface ships. Instead, MEDMOD simulates generic Red surface ships which can consist of up to eight different specific types of ships (as currently dimensioned). Almost all of the effectiveness parameters associated with Red surface ships are a function of the type of Red ship. Red surface ships can simultaneously be in any (or all) of the geographic regions except for Region 0, and there are no computer limitations on the number of Red ships of any type in any region (other than Region 0).

b. Submarines

MEDMOD simulates two types of Red submarines "in regions." These two types of Red submarines can be in any region, except

Region 0. These two types of Red submarines are: (1) submarines that fire (only) torpedoes at the task force, and (2) submarines that fire (only) antiship cruise missiles at the task force. MEDMOD also simulates a third type of Red submarine "in barriers." This type (and only this type) of submarine can form the Red submarine barriers between consecutively numbered regions. There are no computer limitations on the numbers of these submarines that can be in any region (for the first two types) or in any barrier (for the third type).

c. Land-Based Aircraft

MEDMOD can simulate three general classes of Red land-based aircraft: bombers, multiple role fighters (that can fly either escort or intercept missions), and interceptors (i.e., fighters that can only fly intercept missions). MEDMOD can simulate only one type of multiple role Red fighter and only one type of Red interceptor in any one run. However, MEDMOD can simultaneously simulate multiple types of Red bombers in the same run--it is currently dimensioned to hold up to three different types of Red bombers. The bombers and multiple role fighters can be based on either of two notional aggregated airbases, each of which can be thought of as consisting of a (possibly different) number of identical typical airbases. One of these aggregated airbases is assumed to be vulnerable to attack from carrier-based aircraft, the other is assumed to be invulnerable to attack. (MEDMOD simulates only carrier-based air attacks on Red airbases, not ground-based air attacks.) If desired, all Red bombers and fighters could be located on one base or the other, or these aircraft can be split in any manner between these bases. In addition to vulnerability, sortie rates can be a function of base. However, once in the air, the effectiveness parameters of these aircraft are not functions of their home base.

Since one airbase is invulnerable to attack, all multiple role fighters on this airbase always fly escort missions. Multiple role fighters on the other airbase can be split in any manner (by input) between escort and intercept missions. Also, since one airbase is invulnerable to attack, there is no point in stationing interceptors on that base. Thus, interceptors can only be stationed on the notional vulnerable Red airbase in MEDMOD.

Red bombers have only one mission in MEDMOD--to attack the Blue task force. Red fighters on the vulnerable Red airbase can fly escort missions for these bombers (in order to protect bombers from Blue land-based interceptors and carrier-based fighters) or can fly intercept missions (to protect the notional vulnerable Red airbase from Blue carrier-based air attacks). As stated above, Red fighters on the invulnerable Red airbase can only fly escort missions and Red interceptors (which must be on the vulnerable Red airbase) can only fly intercept missions.

d. <u>Ground Defenses</u>

MEDMOD simulates two classes of ground defenses for the vulnerable Red airbase: shelters and surface-to-air missile systems--SAMs. MEDMOD plays only one type of shelter for Red aircraft, but it contains quite flexible rules concerning which types of Red aircraft can and cannot be sheltered. MEDMOD can simulate multiple different types of SAMs defending Red airbases--it is currently dimensioned to hold up to two different types of SAMs.

MEDMOD also simulates Red SAMs defending against attacks by Blue power projection missions. Multiple types of Red SAMs can be played. MEDMOD is currently dimensioned to hold up to two different types of SAMs defending against power projection missions, and these types can be different than the types of SAMs defending Red airbases.

3. Movement of Resources

As indicated above, the Blue task force (and so all resources that are part of the task force, as well as submarines in direct support of the task force) can be automatically moved from region to region during the war simulated by MEDMOD. (This movement is calculated by Subroutine MOVTF in MEDMOD.) Red surface ships and submarines in regions can also be automatically moved from region to region by MEDMOD. (This movement is calculated by Subroutine MOVRS in MEDMOD.) Other than these movements, no resources are automatically moved by MEDMOD, no resources are automatically added to the theater (as reinforcements or replacements) during the war and, except for attrition, no resources are automatically deleted or moved out of the theater during the war.

The term "automatically" here means "according to a set of rules other than direct user input specification that a certain number of a certain type of resource is to be moved from place a to place b at time t no matter what else is happening in the model." MEDMOD always allows any resource to be added, deleted, or moved at the start of any time period by direct input specification (of course, a resource should be moved or deleted at the beginning of a time period only if it has not yet been destroyed by enemy forces).

For example, three additional AAW escorts can be added to the simulation at the start of time period eight by simply stating in the inputs to MEDMOD that the number of escorts is to be incremented by +3 at the start of time period eight. Also, ten Red fighters on the vulnerable airbase can be deleted from the simulation at the start of time period six by stating in the inputs to MEDMOD that the number of Red fighters on the vulnerable airbase is to be incremented by -10 at the start of time period six. Note, however, that a previous run with (essentially) exactly the same inputs must

have been made to determine that there will be at least ten Red fighters "alive" on the vulnerable airbase at the start of time period six (otherwise, the number of these Red fighters would become negative). Similarly, for example, four Red submarines can be moved from the barrier between Regions 0 and 1 to the barrier between Regions 2 and 3 at the start of time period five by stating in the inputs that the number of Red submarines in the first barrier is to be incremented by -4 and the number of Red submarines in the latter barrier is to be incremented by +4 at the start of time period five. Again, a previous run of MEDMOD with essentially the same inputs must have been made to determine that there are at least four Red submarines still in the first barrier at the start of time period five. Note also that if a user of MEDMOD desires this move to take place over, say, three time periods, that during these three time periods the Red submarines will only be moving (they will not be causing attrition to any Blue resources) and that the user believes that, on average, one-half of a submarine will be lost on this move, then this can be done in MEDMOD stating in the inputs that the first barrier is to be incremented by -4 at the start of time period five and that the latter barrier is to be incremented by +3.5 at the start of time period eight.

Resources are incremented using the TIMET subroutine in MEDMOD. This subroutine also allows any value for any effectiveness parameters to be replaced (not incremented) by any other value at the start of any time period.

E. OVERVIEW OF COMBAT INTERACTIONS

In general, combat in MEDMOD consists of a "D-day shoot-out" that is simulated (at most) once at the very start of combat, and of repeated cycles through all other combat interactions.

The purpose of the D-day shoot-out is to reflect the capability of those Red submarines and surface ships that are trailing the task force (i.e., "tattletales") to inflict damage on the task force. A D-day shoot-out cannot occur if the task force is initially in Region 0, and no Red aircraft are involved in this shoot-out.

After the D-day shoot-out, the (all other) combat interaction section is entered. This section consists of cycles through the following interactions or events.

First, a Red air attack against the Blue task force can be generated (no combat occurs here). Second, combat between this Red air attack and the Blue land-based air barrier is simulated (provided that there is a Red attack and that Blue has a land-based air barrier). Third, combat between Red submarines that are attempting to attack the task force and Blue submarines that are in direct support of the task force is simulated. (One can think of this submarine combat as occurring before or during the generation and flight of the Red air attack; the relevant point here is that this submarine combat must be simulated before the Red submarines attack the task force, which comes next.) Fourth, the Red aircraft and submarine attack on the task force is simulated. (This attack is simulated in Subroutine CTFMOD of MEDMOD which, as stated above, is strongly based on the R-245 model of an air and submarine attack on a task force.) Fifth, combat between Red surface ships that are attempting to attack the task force and the task force ships (and aircraft, if the task force contains aircraft carriers) is simulated. Sixth, if there are carriers in the task force, their aircraft can attempt to fly power projection missions.

If the task force has no remaining effectiveness at this point, then no more combat is simulated (the definition of effectiveness used here will be given in Chapter II).

Otherwise, if the rules governing movement of the task force indicate that a move is called for, then the task force is moved to a new region at this point. If such a move is called for and if it involves crossing a Red submarine barrier, then combat between the Red submarines in that barrier and the Blue task force (including submarines in direct support of the task force) is simulated. Red ships and submarines in regions are then to be moved according to their movement rules; and if this movement results in these Red resources crossing Blue submarine barriers, then combat is simulated between these Red resources and the Blue submarines in the barriers being crossed. Finally, an attack by Blue carrier-based fighters and attack aircraft on the notional vulnerable Red airbase is simulated. After this airbase attack, if the input maximum number of time periods has not yet been simulated (and if the task force is not at zero effectiveness) then MEDMOD cycles through these same combat interactions (in the same order) for the next time period. Otherwise, no more combat is simulated, summary results are printed, and the simulation ends.

All of these combat interactions are discussed in greater detail in Chapter II.

F. STATUS OF MEDMOD

This documentation is preliminary in that portions of Chapter II will be discussed in greater detail in the near future in order to more thoroughly document MEDMOD (which will then be called NAVMOD). Very few additional resources or combat interactions will be added to MEDMOD in the near future. Thus, the overview of MEDMOD given in Sections A through E above is (for the near future) virtually final, not preliminary.

In particular, the current status of MEDMOD is as follows: The programming of MEDMOD is complete. That is, an input routine, all of the combat interaction routines, two types of output displays, and the code to hold it all together have been programmed. Nothing essential is missing, but some improvements (primarily in terms of improved output formats and additional output tables) will be added in the near future. An unclassified and entirely hypothetical data base has been prepared, MEDMOD has been successfully run with this data base, and brief initial tests of MEDMOD have been completed.

G. COMPUTER STORAGE SPACE AND RUNNING TIME REQUIREMENTS

The current version of MEDMOD is about 26,100 (decimal) words long. However, because it is overlaid, MEDMOD consumes a maximum of about 22,000 words of core. (This overlay structure is not necessary and would be easy to remove; it was incorporated in MEDMOD in anticipation of possible future expansion.) Of the core consumed, about 1,400 words are used to store the inputs in blank COMMON, and about 100 words are used for labeled COMMON storage--the rest is code. MEDMOD contains about 3,300 FORTRAN statements (counting each distinct COMMON statement only once, not each time it appears, counting continued FORTRAN statements as one statement, and not counting COMMENT cards). There are about 1,000 COMMENT cards (including blank spacers) in the MEDMOD code.

The running time of MEDMOD on IDA's CDC-6400 computer is currently about five seconds (for the one-time calculations, such as Overlay INP and Subroutine DDAY) plus 0.5 to 0.8 seconds per time period simulated.

Chapter II

DISCUSSION OF COMBAT INTERACTIONS

A useful and appropriate way to discuss the combat interactions of MEDMOD is in terms of the major subroutines of the MEDMOD computer program. Section A, below, presents the basic structure of this computer program, and Sections B through N briefly discuss the 13 major subroutines currently used in the MEDMOD computer program.

A. STRUCTURE OF THE MEDMOD COMPUTER PROGRAM

1. Structure of the Overlays

The MEDMOD computer program consists of a main overlay, DRIVER, and two overlay programs, INP and MEDMOD.

Overlay INP only reads in and prints out the inputs. INP has been used to read and print inputs in other models (TACWAR [3], IDATAM [4], and IDACASE [5]). Appendix A contains extracts from [4] and [5] which describe in detail how to input data using INP, and these extracts apply to MEDMOD as well as to IDATAM and IDACASE.

It should be noted that INP contains some computer-specific FORTRAN. Changes required to convert INP to machines other than those like IDA's CDC 6400 are also discussed in Appendix A, and INP has been successfully converted to other machines. (Because INP only reads and displays inputs, INP could be replaced by a different input routine, if a user of MEDMOD so desires, without affecting the logic or code of the combat interactions simulated in MEDMOD.)

Overlay MEDMOD contains the simulations of all the interactions being modeled. Thus, any statement about Overlay MEDMOD also usually applies to the whole MEDMOD computer program and vice versa. In those cases where the distinction is important, the terms "Overlay MEDMOD" for the overlay and "MEDMOD computer program" for the whole program will be used.

Before discussing Overlay MEDMOD, a few words about DRIVER are in order. DRIVER performs three functions: It calls Overlay INP to read and display the inputs, it calls Overlay MEDMOD to perform the simulations, and it writes column headings and related information for the summary output table. (Subroutine PRTSUM in Overlay MEDMOD writes one row of results for each time period under these column headings on the summary output table.) This output table is stored on a separate output file (labeled TAPE10), and the operating system (job control language) prints this output file after the MEDMOD computer program has completed the simulation of the war being examined. This structure allows summary results, which are calculated during each time period, to be written onto an output file (so that they do not take up storage space in computer memory)--yet the output file is not printed until the run is over so that the summary outputs all appear at the end of the run, not mixed in with the detailed outputs for each time period.

2. Overlay MEDMOD

Overlay MEDMOD proceeds in the following manner: First, MEDMOD initializes selected parameters (working variables) and records input values of selected resources for later use. Next, MEDMOD calls Subroutine PRTSUM to write one line of results onto the output file TAPE10 . Since no combat has taken place yet, these results give some initial values of the simulation. Next, MEDMOD calls Subroutine DDAY, provided

that the task force is not in Region 0. Subroutine DDAY simulates the D-day shoot-out between the task force ships and Red tattletale ships and submarines in the same region. MEDMOD then calls Subroutine PRTSUM again to write a line of results which summarize the D-day combat. (If the task force is initially in Region 0, then these first two lines of summary results will be identical.) If the task force contains one or more aircraft carriers and if all of its carriers have suffered sufficient damage in the D-day shoot-out that they now are completely ineffective (according to rules based on input parameters that determine the effectiveness of a carrier as a function of the number of hits it has received), then the simulation stops. Otherwise, MEDMOD proceeds to simulate combat for the first time period.

Combat during the time periods (i.e., all combat other than the D-day shoot-out) is simulated using a loop that runs from 1 to MAXTP, where MAXTP is the input which gives the maximum number of time periods to be simulated (fewer than MAXTP time periods can be simulated, as will be described below).

The first subroutine that can be called in this loop is Subroutine TIMET. This subroutine is called if and only if inputs are to be changed at this time in the simulation. TIMET can automatically increment any resources by any amount and can automatically replace the current value of any parameter with any new value. These incremental amounts and new replacement values must be input to MEDMOD as described in Appendix A. Table B-4 (of Appendix B) gives a complete list of those inputs to MEDMOD which are classified as resource inputs, and so any change to any of these inputs is an incremental change. Changes to any input to MEDMOD other than those listed on Table B-4 results in a replacement by the new value, not an increment.

The next six subroutines that can be called by Overlay MEDMOD are: Subroutines GNAATK, PLBAB, SUBSUB, CTFMOD, SHPSHP, and POWERP. These six subroutines are called here if and only if the task force is not in Region 0. Subroutine GNAATK can generate a Red air attack on the task force. Subroutine PLBAB simulates the attempt by the Red air attack (if there is one) to penetrate the Blue land-based air barrier (if there is one). Subroutine SUBSUB simulates combat between Blue submarines in direct support of the task force and Red submarines and surface ships in the same location as the task force. Subroutine CTFMOD simulates the Red air and submarine attack on the task force; it calculates the amount of warning to the task force and simulates combat between all Blue sea-based resources and all Red resources in this attack. Subroutine SHPSHP simulates combat between the task force and Red surface ships in the same region as the task force. Subroutine POWERP calculates the number of power projection sorties that can be flown by carrier-based aircraft during the time period, it simulates combat between these aircraft and Red SAMs protecting against Blue power projection, and it calculates the number of successful Blue power projection sorties flown by type of Blue aircraft.

Subroutine ADDMOE is the next subroutine called by Overlay MEDMOD. (ADDMOE and the remainder of the subroutines listed here are called whether or not the task force is in Region 0.) ADDMOE determines whether to stop the simulation based on the effectiveness of the task force. If the task force has one or more aircraft carriers, then the simulation is terminated if the carriers have essentially no effectiveness. (Specifically, the simulation is terminated if the input number of carriers, XPLAT, is greater than zero and the working variable XEFFCM, which gives the average effectiveness of the carriers in the task force, is less than 0.00005.) The simulation is also terminated if there are

no Blue surface ships remaining in the task force (which is the only way that the task force can lose all of its effectiveness if the task force does not contain any aircraft carriers).

The next three subroutines that can be called by Overlay MEDMOD are: Subroutines MOVTF, MOVRS, and ABATCK. These three subroutines are called here if and only if ADDMOE has determined that the simulation should not be stopped based on the effectiveness of the task force. (If ADDMOE determines that the simulation should be stopped for this reason, MEDMOD proceeds to call Subroutines PRTRES and PRTSUM as described below.) Subroutine MOVTF determines whether the task force is to be moved to a new region and, if so, it makes this move and simulates combat that occurs between the task force and the appropriate Red submarine barrier (if there is one). Subroutine MOVRS determines the number of Red ships and submarines in each region that are to be moved to another region, it makes these moves, and it simulates combat between these Red resources and the appropriate Blue submarine barriers. Subroutine ABATCK determines whether Blue will make a carrier-based air attack on the notional vulnerable Red airbase (based on the new location of the task force) and, if so, it models this attack.

Subroutines PRTRES and PRTSUM are called after Subroutine ADDMOE if Subroutines MOVTF, MOVRS, and ABATCK are not called, and are called after Subroutine ABATCK otherwise. Subroutine PRTRES is not currently coded; when it is coded, this subroutine will print out the current values of all resource variables in tabular form. Subroutine PRTSUM totals some summary results and writes all the summary results onto an output file (TAPE10).

If either the stopping criterion of no residual effectiveness in the task force has been met (ISTOP = 1) or the

maximum number of time periods to be simulated now has been simulated, the main loop of Overlay MEDMOD is exited, some variables for output displays are calculated, the MEDMOD computer program is ended, and control returns to the operating system level to print the summary output table. Otherwise, the index of time period, ITP, is incremented by 1 and the next time period starts (by calling Subroutine TIMET if inputs are to be changed at the start of this next time period).

A flowchart of the MEDMOD computer program is given in Figure 2. This flowchart is generally correct (at its level of detail) but, for clarity, it omits a few steps like the initialization of selected parameters, the calls to PRTSUM on either side of Subroutine DDAY, and the call to the not yet coded Subroutine PRTRES.

3. Some Assumptions and Conventions

Before discussing the interactions simulated in the subroutines called by Overlay MEDMOD, it is useful to state some simplifying assumptions and conventions used in MEDMOD.

a. Attrition

Most (but not all) of the resources simulated in MEDMOD can be destroyed by enemy resources. The exceptions to this rule are: Blue aircraft carriers, Blue land and carrier-based AEW aircraft, and Blue land and carrier-based ASW aircraft.

Instead of destroying Blue aircraft carriers, MEDMOD lowers their average effectiveness. In particular, XPLAT is the input which gives the initial number of aircraft carriers in the task force, and XEFFCM is a working variable that is initially set equal to 1.0. MEDMOD does not (automatically) change XPLAT. Instead, if the aircraft carriers in the task force have suffered sufficient damage that they are, say,

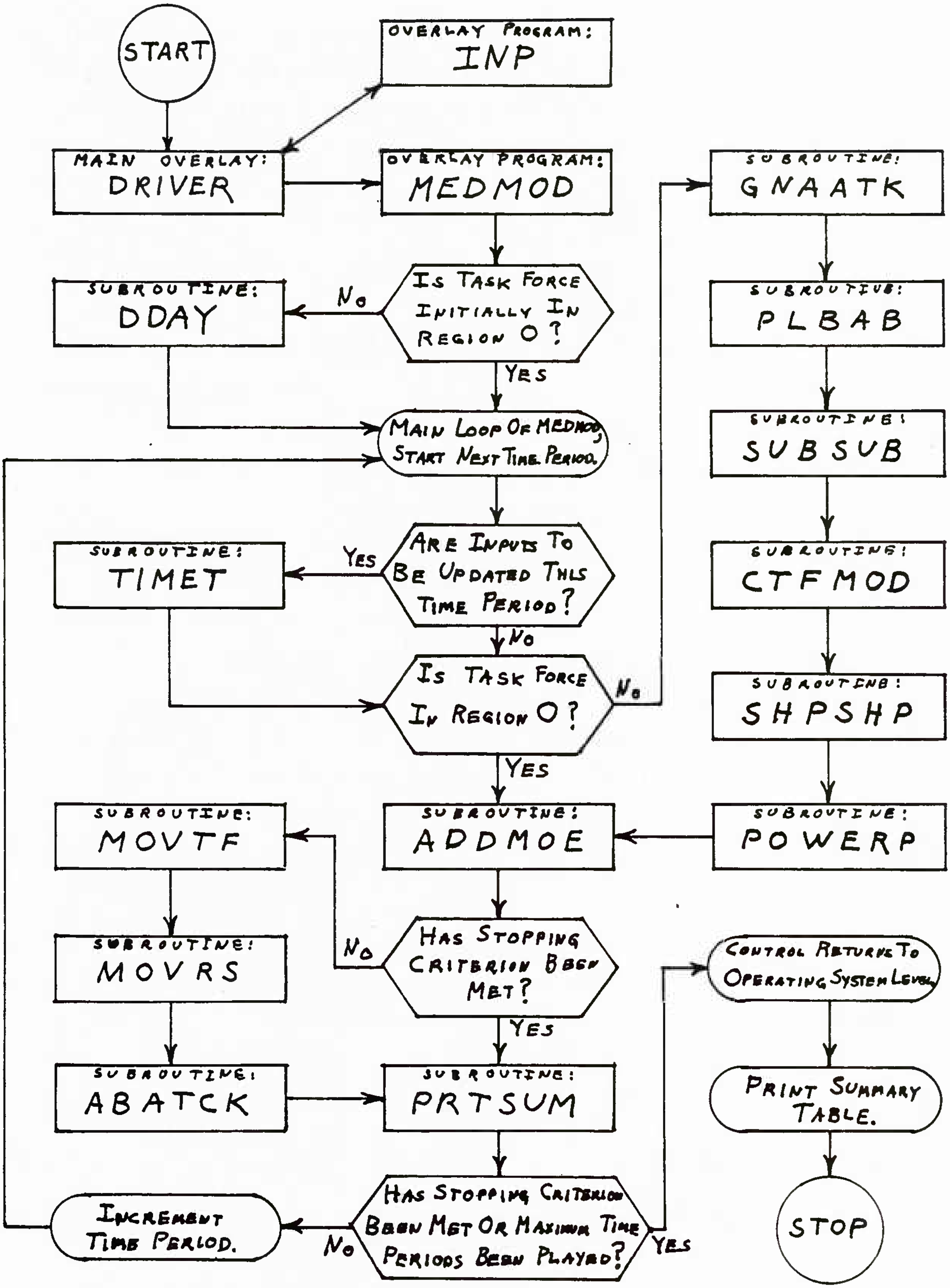


Figure 2. A GENERAL FLOWCHART OF THE MEDMOD COMPUTER PROGRAM

at 80 percent of their initial effectiveness, then XEFFCM will have been reduced to 0.80 in MEDMOD.[1] This convention is reasonable (given the small number of carriers and the deterministic nature of MEDMOD), and is consistent with the R-245 model.

Attrition to AEW and ASW aircraft in the air (caused, perhaps, by Red fighters) is not simulated in MEDMOD because this interaction is not believed to be a significant aspect of the overall combat being modeled. No Red attacks on Blue land bases are simulated, and so attrition to land-based AEW and ASW aircraft on the ground is not modeled. Attrition to carrier-based AEW and ASW aircraft on the carriers is also not simulated, but unlike the other interactions discussed above, this attrition will be simulated in MEDMOD in the near future. However, it should be noted that, as the effectiveness of the carrier (measured by XEFFCM) decreases, the number of sorties that can be flown by carrier-based ASW aircraft decreases. Thus, attacks on the carrier degrade ASW capability only (as the program stands now) by degrading the carrier's capability to launch ASW sorties, not (yet) by directly killing ASW aircraft on the carrier. (Carrier-based AEW aircraft have a constant sortie rate as the program stands now--in the near future this will also be changed.)

In summary, the number of carriers, XPLAT, is constant in MEDMOD, but the average relative effectiveness of these carriers, XEFFCM, is degraded as the carriers suffer damage. The numbers of land-based AEW aircraft associated with each region L, XAEW(L), and the number of carrier-based AEW aircraft, XAEW, are constant in MEDMOD and their sortie rates are not degraded; however, attrition to fighter aircraft and

---

[1]For example, the number of Blue attack aircraft available to attack Red airbases during any time period either is the number of attack aircraft "alive" on the carrier or is XEFFCM times the initial number of attack aircraft on the carriers, whichever is less.

reduction in their sortie rate due to carrier damage reflects a loss in capability to use this early warning. The number of land-based and carrier-based ASW aircraft, XASWLQ(L) and XASW, respectively, are also constant in MEDMOD. The code will be changed soon to allow attrition to carrier-based AEW and ASW aircraft. Other than these exceptions, all resources simulated in MEDMOD can suffer attrition due to attacks by appropriate enemy resources.

b. Time Periods and Clock Time Periods

Time periods in MEDMOD correspond to one cycle through the main loop in Overlay MEDMOD. Thus, after TIMET, the first calculation made in a time period is to attempt to generate a Red air attack on the task force (Subroutine GNAATK), and the last interaction simulated is the Blue air attack of the notional vulnerable Red airbase (Subroutine ABATCK). MEDMOD places no formal restrictions on the length of a time period. (MEDMOD makes no attempt to distinguish daylight from nighttime.) For some purposes, it can be convenient to think of a time period as being one day long. If this is done, the day should not be thought of as starting with Subroutine GNAATK the first thing in the morning and ending with Subroutine ABATCK at night. Instead, the day should be thought of as starting with Blue (possibly) attacking Red airbases (Subroutine ABATCK), then Red (possibly) generating an air attack on the task force (Subroutine GNAATK), and so on. Blue power projection might be considered as being flown throughout the day, and forces could be moved (Subroutines MOVTF and MOVRS) overnight. As the next morning dawns, Blue could launch the next day's attack (if there is one) against Red airbases.

In order to distinguish a "day," which runs from Subroutine ABATCK through the movement subroutines, from a cycle,

which runs through the loop in Overlay MEDMOD, the following terminology will be used. The period starting with Subroutine ABATCK and running through the movement subroutines will be called a "clock time period," while the cycle through the loop in Overlay MEDMOD will (continue to) be called a "time period." This distinction is formally used in MEDMOD only in one respect: If Blue aircraft can only fly a certain number of sorties per day (or per clock time period), then the day (or clock time period) in MEDMOD starts at Subroutine ABATCK, and ends (for sortie flying purposes) with POWERP--it does not run from GNAATK to ABATCK.

4. The Major Subroutines of MEDMOD

Overlay MEDMOD calls 14 subroutines. One of these, Subroutine PRTRES, has not yet been coded. (Subroutine PRTRES, when coded, will only print out current numbers of resources, by type and by location; it will not perform any simulation-related calculations.) The remaining 13 subroutines are called the major subroutines of MEDMOD.[1] These 13 major subroutines are the 13 that appear in Figure 2, namely: DDAY, TIMET, GNAATK, PLBAB, SUBSUB, CTFMOD, SHPSHP, POWERP, ADDMOE, MOVTF, MOVRS, ABATCK, and PRTSUM. The order here is, with one exception, the order in which these subroutines would be called by MEDMOD if all were called as soon as possible. The one exception is PRTSUM, which is called before and after DDAY, but whose main use is to assist in displaying results at the end of each time period as shown in Figure 2; therefore, PRTSUM follows ABATCK in the order used in this discussion.

These 13 major subroutines are described in the next 13 sections of this chapter (one section for each subroutine). The order of these sections matches the ordering of the

[1]MEDMOD contains several subroutines in addition to these 13 major subroutines; these other subroutines are called by these 13 subroutines (or by each other), not by Overlay MEDMOD.

subroutine stated just above. Each of these sections (except for the section for TIMET) contains a table which lists inputs (parameters and resource variables) used in the corresponding subroutine.[1] (Subroutine TIMET does not use the value of any input variable.) Definitions of these inputs are given in Appendix C. (All inputs to MEDMOD are stored in blank COMMON.) In addition to listing the inputs, each of these tables also lists the major indexing variables used in the corresponding subroutine, and these indexing variables are defined and discussed in Table B-6 of Appendix B. Finally, each section lists the variables stored in labeled COMMON blocks that are used by the corresponding subroutine. Definitions of these (necessarily working) variables are given in Table B-3 of Appendix B. Thus, Tables B-3 and B-6 of Appendix B and all of Appendix C may be quite useful when reading the following sections. Appendix E contains a copy of the MEDMOD code, which can be consulted for specific details concerning these 13 major subroutines (or concerning any portion of the MEDMOD computer program).

Each major subroutine (except for TIMET) prints out selected results of its simulation and calculations each time it is called. Thus, the output of a MEDMOD run currently consists of the following displays. First, INP prints out the inputs; a sample output of INP is given in Appendix C. Next, except for TIMET, each of the major subroutines described below (and some of the subroutines called by these major subroutines) prints out its results whenever it is called. This output, in total, is called the "detailed output" of MEDMOD, and a sample detailed output is given in Appendix D. (Accordingly, a brief referral to Appendix D can also be useful when reading the following sections.) Finally, the summary results are printed--a sample of this summary is given at the end of Appendix D.

---

[1]These tables list all inputs used, except those that are index limits. See Table B-7 for information on these input limits. Definitions of resource inputs are also given in Table B-4.

B. SUBROUTINE DDAY

Subroutine DDAY models the D-day shoot-out. It is called once, at the beginning of the simulation, and then only if the task force is not initially in Region 0. It computes attrition to Blue surface ships in the task force, to aircraft on carriers in the task force, and to Red surface ships and submarines that are in the same region as the task force.

Table 1 lists inputs used in Subroutine DDAY(L).

Table 1. Inputs and Major Indexing Variables Used in Subroutine DDAY(L)

```
INPUT PARAMETERS USED IN SUBROUTINE DDAY
    DDFAC(KRS)              DDSPA(KRS)
    DDPKC(KRS)             *ENACDS(KRS)
    DDPKS(KRS)             *ENACDT(NKRBP1)
    DDRKAA(KRS)             IDDAC
    DDRKBA(KRS)             IDDAS
    DDRSA(KRS)
   *THIS VARIABLE IS ALSO USED IN ANOTHER SUBROUTINE

RESOURCE VARIABLES USED IN SUBROUTINE DDAY
    XPLAT(W/ XEFFCM)        XATTCK
    XEAAW                   XFGHTP
    XEASWA                  RS(1,L)
    XEASWN                  RS(2,L)
    XURGS                   RS(KRS,L),KRS≥3

INDEXING VARIABLES USED IN SUBROUTINE DDAY
    KRS
    L
    (NKRBP1)
```

The subroutine parameter L is the <u>initial</u> location of the task force, which is given by the input LTFMP(1). (See the description of Subroutine MOVTF for further discussion of LTFMP.) Subroutine DDAY makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
ATTCKI	--	COMCTF
FGHTRI	--	COMCTF
XEFFCM	--	COMCTF

The central interaction in DDAY is an attack by Red ships on the task force. The carriers can have different vulnerability than the other Blue surface ships in this attack, but all Blue non-carrier ships are assumed to be equally vulnerable as targets. (Blue direct support submarines are assumed not to be involved in D-day combat.)

Initially there are RS(KRS,L) Red ships of kind KRS in the region. (KRS varies from 1 through NKRS.) The input DDRKBA(KRS) gives the number (not the fraction) of these that are killed before the attack on the task force. (Throughout this subroutine, checks are employed to ensure that no resource variable drops below zero.) The number of Red ships of type KRS that attack the task force on D-day is given by the minimum of the number of Red-type KRS ships that remain and the input DDRSA(KRS). Of these attacking Red ships, an input fraction (given by DDFAC(KRS)) attack carriers; the rest attack other Blue surface ships. A Red ship of kind KRS fires an input number, DDSPA(KRS), of shots. These shots need not all be fired at the same target.

There are two different methods available to calculate attrition to Blue. Method 1 assumes that Red shooters are independent of one another; the attrition equation is the same as that of Subroutine BINOAT, with NKRS kinds of shooters and one kind of target. (See the BINOAT code and [6] for details.) Method 2 assumes perfect coordination between the different Red shooters, so that shots are distributed as evenly over the targets as possible. In both methods, each shot by a ship of kind KRS aimed at a carrier has (input) probability of

DDPKC(KRS) of killing it;[1] a shot aimed at a Blue non-carrier surface ship has probability of kill DDPKS(KRS). The input variables IDDAC and IDDAS determine the method used for computing attrition to carriers and attrition to other Blue surface ships, respectively.

Red ships do not suffer attrition <u>during</u> the attack on Blue. <u>After</u> the attack the input number (not fraction) DDRKAA(KRS) of the Red ships of kind KRS (1 through NKRS) are killed.

C. SUBROUTINE TIMET

The purpose of Subroutine TIMET has been discussed at some length above, and the procedure for changing input values using TIMET is discussed in Appendix A--see, in particular, Section II.C of that appendix. This section will supplement those discussions by making three comments about the specific use of TIMET in MEDMOD and one general comment about incremental variables which apply to any model that uses TIMET.

The three comments about the use of TIMET in MEDMOD are as follows. First, the parameter of Subroutine TIMET(ICYCLE), namely ICYCLE, corresponds to the current time period in Overlay MEDMOD. Furthermore, when TIMET is called, it is called at the start of each time period. Thus, if a user of MEDMOD desires to change the value of an input at the start of time period 11, for example, 11 should be entered in columns 19 and 20 of the appropriate data card as described in Section II.C of Appendix A. (This relationship is different than that used, for example, in IDATAM. IDATAM calls TIMET at the end of its periods; so if a variable is to have a different value in period 11 than it had in period 10 in IDATAM, 10 (not 11) should be entered in columns 19 and 20

[1]Carriers are not actually killed. Their fighter and attack aircraft (variables XFGHTR and XATTCK) are killed and their average effectiveness (variable XEFFCM) is degraded.

of the appropriate IDATAM data card.) TIMET contains one labeled COMMON block, COMIGO, and this COMMON block contains one variable, IGO. The next time period in which changes are to be made is stored in IGO, and Overlay MEDMOD calls Subroutine TIMET at the start of a time period only if IGO indicates that changes are to be made then (i.e., only if the index of time period ITP equals IGO).

Second, since TIMET makes changes at the beginning of the loop in Overlay MEDMOD, entering 01 (or [blank]1) in columns 19 and 20 of an input data card would have the same effect as changing the initial inputs--except for inputs used in Subroutine DDAY. Changes made at the start of time period 1 to those inputs used only in Subroutine DDAY would have no impact on the course of the simulation, and changes at the start of time period 1 to those inputs used both in DDAY and in other major subroutines of MEDMOD would be applied after DDAY is called but before those other subroutines are called.

Third, TIMET either increments the current value of an input or it replaces the value of an input. Each input is treated in only one way--either it is always changed by incrementing its value or it is always changed by replacing its value. In MEDMOD the resource inputs listed on Table B-4 of Appendix B are all incremental inputs, and these inputs are the only incremental inputs; all other inputs to MEDMOD are replacement inputs.

A general comment concerning TIMET is as follows. TIMET inputs must be grouped by cycle (i.e., time period) and these groups must be in ascending order by cycle number. The ordering of different inputs within the same cycle is irrelevant. If an incremental input is incremented multiple times in the same cycle then all of these increments apply (in a cumulative manner), not just the last increment. Of course, if a replacement variable is given several new values in the same cycle,

then only the last value is used. And, as described in Appendix A, if there are multiple <u>initial</u> data cards for <u>any</u> input, then only the last value is used.

Finally, TIMET does not make use of the value of any input--it only changes these values. TIMET references the input NEPD, but it uses NEPD only to determine the addresses of other entries in blank COMMON (i.e., all other inputs); it does not use the value of the (dummy) input NEPD.

D. SUBROUTINE GNAATK

Subroutine GNAATK determines whether Red will launch an air attack against the task force during the current time period and, if so, it determines what the size and composition of this attack will be.

Table 2 lists inputs used in Subroutine GNAATK(L,ITP).

Table 2. Inputs and Major Indexing Variables Used in Subroutine GNAATK(L,ITP)

```
INPUT PARAMETERS USED IN SUBROUTINE GNAATK
    AVAILE(L,IAB)
    AVAILT(L,IAB,KRB)
    BMTMIN(L)
    IATKRT(L)

RESOURCE VARIABLES USED IN SUBROUTINE GNAATK
    ATABT(IAB,KRB)
    AESCAB(IAB)

INDEXING VARIABLES USED IN SUBROUTINE GNAATK
    IAB
    KRB
    L
```

The subroutine parameter L in the current location of the task force, and the parameter ITP gives the current time period. Subroutine GNAATK makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
BMR	--	COMGA
ESC	--	COMGA
NTPSLA	--	COMGA

If the number of time periods since the last Red air attack on the task force (given by NTPSLA) is greater than zero but less than the input IATKRT(L), then no Red air attack is generated, NTPSLA is incremented by one, and the subroutine ends. NTPSLA is zero only if there has not yet been an air attack on the task force. Thus, if NTPSLA is zero or is greater than or equal to IATKRT(L), Red can attack (if enough Red bombers are available). The number of available bombers and escorts are calculated next. If the total number of available bombers is less than the input BMTMIN(L), there is no attack. Otherwise, there will be an attack consisting of BMR(IAB,KRB) Red bombers of type KRB from Red airbase IAB, and ESC(IAB) Red escorts for Red airbase IAB, where IAB = 1 denotes the notional vulnerable Red airbase, and IAB = 2 denotes the notional invulnerable Red airbase. If there is an attack, NTPSLA is set to one.

After these calculations, Subroutine GNAATK prints out the numbers of Red aircraft (by type and airbase) in the attack, the numbers remaining on the airbases, the new value of NTPSLA, and it ends--no attrition is simulated in GNAATK.

E. SUBROUTINE PLBAB

Subroutine PLBAB simulates the attempt by the Red air attack generated in Subroutine GNAATK (if such an attack was generated there) to penetrate the Blue land-based air barriers (if the inputs state that there is such an air barrier).

Table 3 lists inputs used in Subroutine PLBAB(L). The The subroutine parameter L is the current location (or region)

Table 3. Inputs and Major Indexing Variables Used in Subroutine PLBAB(L)

```
INPUT PARAMETERS USED IN SUBROUTINE PLBAB
    IPLACA               PLPDDA(KBD)
    IPLAED               PLPDDE(KBD)
    PLAEDA(KBD)          PLPDED
    PLAEDE(KBD)          PLPKAD(KRA,KBD)
    PLAEED               PLPKDA(KBD,KRA)
    PLCA(L)              PLPKDE(KBD)
    PLFDLL(LB,L,KBD)     PLPKED(KBD)
    PLPAJO(KRA)

RESOURCE VARIABLES USED IN SUBROUTINE PLBAB
    PLBLBD(KBD,LB)
    ATABT(IAB,KRB)
    AESCAB(IAB)

INDEXING VARIABLES USED IN SUBROUTINE PLBAB
    IAB                  KRB
    KBD                  L
    KRA                  LB
```

of the task force. Subroutine PLBAB makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
BMR		COMGA
ESC		COMGA

First, Subroutine PLBAB determines whether there is a Red air attack or not. If not, the subroutine ends; if there is one, it continues.

Next, the subroutine determines the number of Blue land-based interceptors of type KBD, D(KBD), that form the Blue land-based air barrier. D(KBD) is computed by

$$D(KBD) = \sum_{LB=1}^{NLOC} PLBLBD(KBD,LB) * PLFDLL(LB,L,KBD) ,$$

where PLBLBD(KBD,LB) is the number of Blue type-KBD interceptors associated with Region LB, and PLFDLL(LB,L,KBD) is the fraction of these interceptors that fly sorties to participate in the land-based air barrier given that the task force is in Region L.

Subroutine PLBAB then calls Subroutine AIRAIR which simulates and computes attrition for the following two interactions in the following order:

1. Blue interceptors versus Red escorts.
2. Surviving Blue interceptors versus Red bombers.

Most of the input parameters starting with "PL" are detection or kill probabilities or other parameters for these interactions. Subroutine AIRAIR is also called by Subroutine ABATCK, as stated in Section M, below; however, a description of Subroutine AIRAIR will not be given in this preliminary documentation.

Results of the first interaction are: the numbers of Red escorts and Blue interceptors (by type) that are alive and continuing on their mission, the numbers of these aircraft that are alive but are unable to participate in additional engagements (e.g., due to lack of ordnance) and so are returning to their home base, and the numbers of these aircraft that are killed. The results of the second interaction are: the numbers of Red bombers (by type) that are alive and continuing on their mission, the numbers of these bombers that are alive but have jettisoned their ordnance in this interaction and so are returning to their home base, the numbers of these bombers that are killed, the numbers of Blue interceptors (by type) that are alive and returning to their home base, and the numbers of these interceptors that are killed. (Note that Blue interceptors can be killed by Red bombers in this interaction if the inputs to MEDMOD state that some types of Red bombers are carrying both air-to-air and

air-to-ground ordnance and that these types of bombers can return fire against Blue interceptors after jettisoning their air-to-ground ordnance. Note also that since this interaction completes the mission of these Blue interceptors, all Blue interceptors that survive return home here.)

The results of these two interactions can depend on the numbers and types of aircraft involved, but not on which bases these aircraft are stationed. The results applying to each particular type of aircraft are distributed over the aircraft of that type from each airbase according to the proportion of aircraft of that type participating from that airbase. Of course, if there are no Blue interceptors of any type from any airbase, then no kills occur and all Red aircraft continue on their mission.

Subroutine PLBAB ends after printing its results.

F. SUBROUTINE SUBSUB

Subroutine SUBSUB models two different interactions each time it is called. In particular, it models Blue direct-support submarines versus Red submarines and it models Blue direct-support submarines versus Red surface ships. The first interaction is intended to model an advance screen of Blue submarines trying to prevent Red submarines from reaching the task force, and so SUBSUB is called before Subroutine CTFMOD in Overlay MEDMOD.

Table 4 lists inputs used in Subroutine SUBSUB(L). The subroutine parameter L is the current location of the task force. This subroutine does not use any labeled COMMON blocks.

The number of Blue submarines present is BSSNDS. (Except for barrier submarines, all Blue submarines travel with the task force.) There are RS(1,L) Red torpedo submarines and RS(2,L) Red missile submarines in the region, but only the

input fraction SBFRSA(L) of them are assumed to be able to participate in the interaction. Furthermore, only the input fraction SBFRSC of these are assumed capable of shooting at Blue. The input fraction SBFBCS of the Blue submarines are assumed capable of shooting at Red submarines. (The two kinds of Red submarines are indistinguishable in this subroutine; the different kinds of Red surface ships are also treated as identical. Participation and attrition are assessed proportionately over the kinds of Red ships. These simplifying assumptions might be changed in the near future.)

Table 4. Inputs and Major Indexing Variables Used in Subroutine SUBSUB(L)

```
INPUT PARAMETERS USED IN SUBROUTINE SUBSUB
    SBFBCF                SBPBDS
    SBFBCS                SBPBKF
    SBFRFA(L)             SBPBKS
    SBFRFC                SBPFDB
    SBFRSA(L)             SBPFKB
    SBFRSC                SBPSDB
    SBPBDF                SBPSKB

RESOURCE VARIABLES USED IN SUBROUTINE SUBSUB
    BSSNDS
    RS(1,L)
    RS(2,L)
    RS(KRS,L),KRS≥3

INDEXING VARIABLES USED IN SUBROUTINE SUBSUB
    KRS
    L
```

Attrition is computed by Subroutine BINOAT; the code for this subroutine contains (in comment cards) a description of the attrition equation used. For more background on this attrition equation see IDA Paper P-1031 [6].

Subroutine BINOAT only computes attrition to targets caused by shooters. To calculate attrition to both sides,

a "shoot-then-shoot-back" scheme is used in SUBSUB. This is performed using the variables:

BSA = Blue submarines present (i.e., BSSNDS),

BSCS = Blue submarines capable of shooting,

RSA = Red submarines present, and

RSC = Red submarines capable of shooting.

First, BINOAT is called with BSCS shooters and RSA targets resulting in RSK1 targets killed. The surviving (RSA-RSK1)*SBFRSC capable Red submarines then attack the BSA Blue submarines, resulting in BSK1 submarines killed. The procedure is then reversed. BINOAT is called with RSC shooters and BSA targets; BSK2 targets are killed. The surviving (BSA-BSK2)*SBFBCS Blue submarines now attack RSA Red targets, killing RSK2.

The overall Blue attrition is computed as the weighted average

$$BSK = \left(\frac{BSCS}{BSCS+RSC}\right)BSK1 + \left(\frac{RSC}{BSCS+RSC}\right)BSK2 \quad .$$

The overall Red submarine attrition, RSK, is a weighted average of RSK1 and RSK2, with the same weights as above.

Rationale that underlies "shoot-then-shoot-back" schemes is discussed in [7] (which also appears as Chapter D.I of [8]) and in [9].

The surviving BSA-BSK Blue submarines then can attack Red surface ships. The input fraction SBFBCF of them are assumed capable of engaging Red surface ships. There are RS(KRS,L) Red ships of kind KRS in the region, where KRS runs from three through NKRS and indexes the kinds of Red surface ships.[1] The input fraction SBFRFA(L) of these Red ships are

---

[1]Note that the input variable NKRS must be at least three, i.e., there must be at least one kind of surface ship, even if (continued on next page)

assumed to participate in the interaction; of these the input fraction SBFRFC are assumed capable of shooting at Blue. First, the capable Blue submarines shoot at the Red surface ships; attrition is assessed by BINOAT. Then the surviving capable Red surface ships shoot back at Blue.

After all the attrition has been assessed, relevant quantities are printed and updated as appropriate.

G. SUBROUTINE CTFMOD

This section consists of two parts. The first part will describe Subroutine CTFMOD in a manner similar to the description of other major subroutines presented here. This description is relatively brief, and it will be expanded or supplemented in a later ("full" rather than "preliminary") documentation of MEDMOD. The second part lists the major differences between Subroutine CTFMOD and the R-245 model. Thus, a reader who is not familiar with IDA Report R-245 [1] can skim or omit the second part; a reader who is familiar with R-245 should read the second part and might do so before reading the first part below.

1. Description

Subroutine CTFMOD models several different interactions. It simulates attacks by Red submarines that fire torpedoes and by Red submarines that fire antiship cruise missiles (ASMs) at the task force. If there is an air attack on the task force, it models the extent of early warning that the task force receives, it simulates combat involving carrier-based CAP and DLI fighters versus Red bombers and Red escort aircraft for these bombers. It models the area-air-defense (AAD) and ship-self-defense (SSD) capabilities of the ships in the task force to destroy bomber-launched and submarine-launched

(cont'd) there are zero ships of that kind. If NKRS < 3, the program stops.

ASMs, and it models the effect on task force ships of the torpedoes and ASMs that penetrate these defenses.

Table 5 lists inputs used in Subroutine CTFMOD(L). The subroutine parameter L is the current location of the task force. Subroutine CTFMOD makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
ATSORU	--	COMSOR
ATTCKI	--	COMCTF
BMR	--	COMGA
ESC	--	COMGA
FGHTRI	--	COMCTF
FTSORU	--	COMSOR
XCAPST	--	COMCTF
XEFFCM	--	COMCTF

The first computation made in Subroutine CTFMOD is to determine whether there is an air attack on the task force during the time period in question. If there is no air attack and if Red submarines (of both types) will attack the task force only during time periods in which there is also an air attack (as determined by the input IRSUBA(L)), then the subroutine ends; otherwise, it continues.

Next, Subroutine CTFMOD initializes certain working variables to appropriate values based on the current location of the task force, and it prints out the values of these (and some related) variables. If Red submarines can attack the task force, then an input fraction, FSTAQ(L), of the torpedo-firing submarines do so. If there is no Red air attack and if ASM submarines only attack the task force during time periods when there is also an air attack (also as determined by IRSUBA(L)), then the number of attacking ASM submarines is set to zero. Otherwise, FSTGAQ(L) of the ASM-firing submarines in the region attack the task force.

Table 5. Inputs and Major Indexing Variables Used in Subroutine CTFMOD(L)

INPUT PARAMETERS USED IN SUBROUTINE CTFMOD

AEWD	HRMASW	SSDASW	VCAP
ASWF	HRMURG	SSDURG	VI
BAREAQ(L)	HRTAAW	STARQ(L)	WFMAAW
BARELQ(L)	HRTASW	STSALV	WFMASW
BARLQ(L)	HRTURG	SUBSOR	WFMPLT
*BUCAP	IATRIA	TAB10T(I,K)	WFMURG
CAPMLQ(L)	IRSUBA(L)	TAB12(I)	WFTAAW
CAPMQ(L)	PDIN	TAB13T(I,K)	WFTASW
CAPMR	*PFFCNF	TCAP	WFTPLT
CAPSTQ(L)	PKASW	THSCAQ(L)	WFTURG
DLIA	PKAT1	THSCTQ(L)	WRLNDQ(L)
D1T(I,KRB)	PKDF1	TPS	WVSIZ
D2T(I,KRB)	PKIIN	T1	ZLAMPF
*ENACDT(K)	PKIN	T2	ZMPATT(K)
ESLR	PKPLDT(K)	T3	ZMPCAP
ESRQ(L)	PKPL1	T4	ZMPDLI
FPPL1	PKPL2	UBAEWL	ZMPESC
FPPL2	PKSST(K)	UBAEW	ZMPSTG
FSTAQ(L)	PRWLNQ(L)	UBASWL	
FSTGAQ(L)	SMALLR	UBASW	
HRMAAW	SSDAAW	VBT(K)	

*THIS VARIABLE IS ALSO USED IN ANOTHER SUBROUTINE

RESOURCE VARIABLES USED IN SUBROUTINE CTFMOD

XPLAT(W/ XEFFCM)	XURGS	XAEWLQ(L)	RS(2,L)
XEAAW	XATTCK	XASW	ATABT(IAB,KRB)
XEASWA	XFGHTR	XASWLQ(L)	AESCAB(IAB)
XEASWN	XAEW	RS(1,L)	

INDEXING VARIABLES USED IN SUBROUTINE CTFMOD

I	K	L
IAB	KRB	

Next, the ASW portion of CTFMOD is simulated. Both torpedo and ASM submarines can suffer attrition due to land-based and carrier-based ASW aircraft. After this attrition is inflicted, ASM submarines fire their missiles and return to the open sea--they suffer no additional attrition in CTFMOD. Torpedo submarines, however, must penetrate through the air barrier imposed by aircraft (helicopters) from the air-capable ASW escorts, and then they must penetrate through the ship barrier imposed by (both types of) ASW escort ships. After penetrating these barriers, each surviving torpedo submarine launches an input number, STSALV, of salvoes of torpedoes (with TPS torpedoes per salvo) at the task force and then returns (with no further attrition) to the open sea. Subroutine CTFMOD uses the number of Blue surface ships in the task force by type, and the input parameters WFTAAW, WFTASW, WFTPLT, and WFTURG, to determine the targeting of these torpedoes on the various types of ships in the task force, and it determines the average number of torpedoes, XSPPLA, targeted against each carrier. Subroutine CTFMOD then uses XSPPLA, the input array TAB12, and the function FUNC12 to determine a factor (XPST) which, when applied to the pre-torpedo attack average effectiveness of the carriers, gives the post-torpedo attack average effectiveness of the carrier.

If there are no missile attacks on the task force (due either to Red bombers or to Red submarines), then XPST is applied to XEFFCM to give a new value for the average relative effectiveness of the carriers, attrition to the other ships in the task force is calculated and applied, these results are printed, and the subroutine ends. Otherwise, if there is a missile attack, XEFFCM is changed and attrition is applied after the number of penetrating missiles has been calculated as described below.

The first step in considering the missile attack is to determine the amount of warning that land and sea-based AEW aircraft provide, to determine the total warning based both on AEW aircraft and on warning due to sensors on the ground (whose effectiveness is input to MEDMOD via the input parameters PRWLNQ(L) and WRLNDQ(L)), and to determine the number of CAP stations that can be supported considering, in part, the current number of fighter aircraft on the carriers and the current level of effectiveness of the aircraft carriers.

Next, CTFMOD enters a loop over certain attrition calculations. In particular, this loop consists of two parts. The first part calculates attrition to Red bombers (by type), Red escorts, Blue CAP fighters, and Blue DLI fighters. The second part calculates attrition to ASMs launched from surviving Red bombers caused by ship-based defenses (i.e., by AAD and SSD systems) and it calculates the loss of carrier capability and the attrition to other Blue surface ships due to penetrating ASMs.

The reason for a loop here is as follows. The first pass through this loop calculates attrition to the aircraft involved that occurs before the Red bombers can launch their ASMs against the task force. The only result used from this calculation is the number of Red bombers (by type) killed before they launch their ASMs--attrition to the other aircraft is ignored in this first pass. This number of Red bombers killed by type is subtracted from the corresponding number of Red bombers entering the engagement to determine the number that launch ASMs, which is used in the second part of the first pass through the loop.[1] The second pass through this loop calculates total attrition (both before and after ASM launch) to all aircraft (i.e., to Red bombers and escorts,

---

[1]Different types of Red bombers can launch different types of ASMs, but all Red bombers of the same type must launch the same (continued on next page)

and to Blue CAP and DLI fighters). This total attrition is assessed against the prebattle numbers of these aircraft--no attrition results are ignored in this second pass through this part of the loop. However, the second pass skips entirely the second part of the loop--there is no reason to repeat these calculations since the results of the ASM versus task force battle have already been computed in the first pass. Of course, procedures other than this double loop could have been used here; this particular procedure was selected to reduce the changes required to an already programmed model.

The only difference between the first pass and the second pass through the first part (i.e., the air-to-air battle) of this loop is that the first pass uses the inputs D1T(I,K) and D2T(I,K) for I=1, while the second pass uses these inputs for I=2. (K denotes the type of Red bomber here.) These inputs are distances from the task force, I=1 gives the distance at which bombers of type K launch their ASMs, I=2 gives the distance at which these bombers turn around and become invulnerable. That is, CTFMOD does not explicitly model the process of the bombers being vulnerable as they turn around or when they are in a tail-chase mode. Instead, CTFMOD requires that each bomber (by type) must penetrate to a certain (input) distance from the task force which is, in general, closer than its ASM release range. It is vulnerable during this additional penetration, but turns around and becomes invulnerable as soon as it reaches this distance from the task force. This input distance should be selected so that the additional vulnerability it creates matches the true additional vulnerability of bombers (by type) turning around and flying away from the task force.

(cont'd) (perhaps averaged) type of ASM. Each surviving Red bomber of type K launches the input number ZMPATT(K) of ASMs of type K.

Details concerning this air-to-air portion of CTFMOD (and concerning Subroutine ATRTIA which is called in this portion of CTFMOD) will be given in a later version of this documentation. (In the meantime, the computer code can be consulted for specifics.)

The second part of this loop in CTFMOD (which is exercised only during the first pass through the loop) computes the results of the ASM versus task force battle. Kills of ASMs due to AAD and SSD systems are extracted, and surviving ASMs impact upon their target ships. AAD capability is computed using, in part, the input arrays TAB10T and PKSST and the functions FUNCT9 and FUNC10. The target ships of the ASMs that penetrate the AAD defenses are calculated using the inputs WFMAAW, WFMASW, WFMURG, and WFMPLT.

The average SSD capability of the carriers is computed using, in part, the inputs FPPL1, and FPPL2, the working variable XEFFCM, and the function FUNCT5. Carriers can also have passive defenses, and this capability is computed using the input PKPLDT(K). CTFMOD thus uses the number of ASMs that penetrate through these SSD defenses and are targeted against the carrier, along with the input array TAB13T and the function FUNC12, to determine the degradation factor to apply to XEFFCM to give the new relative average effectiveness of the carrier after absorbing the hits by these ASMs. This factor for damage due to ASMs is denoted by XPSA in CTFMOD. (Remember, XPST, the factor for damage to carriers due to torpedoes, has already been computed but has not yet been applied.) The new (post-attack) value of XEFFCM is then computed as the product of the old (pre-attack) value of XEFFCM times XPSA times XPST.

SSD defenses of the other ships in the task force are computed using the inputs SSDAAW, SSDASW, and SSDURG. The number of ships destroyed by the ASMs that penetrate these defenses is determined using the inputs PMAAW, PMASW, PMURG,

HRMAAW, HRMASW, and HRMURG. If there is an ASM attack (so that destruction due to torpedoes to these ships has not yet been computed), then the destruction due to torpedoes is determined here using the inputs PTAAW, PTASW, PTURG, HRTAAW, HRTASW, and HRTURG. Based on these inputs, the number of penetrating ASMs and torpedoes, the targeting of these ASMs and torpedoes, and on the pre-attack number of ships, the number of surviving AAW escorts, air-capable ASW escorts, nonair-capable ASM escorts, and URG ships are computed and the (first pass of) the loop ends. (Again, a more detailed description of the ASM versus task force battle will be given in a later version of this documentation. In the meantime, the computer code can be consulted for specifics.)

It should be noted that CTFMOD and the R-245 model assume that all ASMs and torpedoes arrive at their target ships within a relatively short time interval. At first this may sound like a Red-favorable assumption due to possible saturation effects. In one sense, it could be Red-favorable in that if submarine-launched ASMs arrived at a sufficiently different time than bomber-launched ASMs, then, in reality, there would be no saturation assisting ASMs launched from either mode due to ASMs launched from the other launch mode. However, in reality, if some ASMs arrive ahead of others, then the ASMs that arrived first could damage the defenses of the task force which would make it easier for the ASMs that arrive later to penetrate. This interaction is not simulated in CTFMOD or in the R-245 model--all arriving ASMs face the pre-attack defenses of the task force because all ship damage is assessed at the end of the whole attack. Also, CTFMOD and the R-245 model do not inhibit the task force from killing torpedoes due to the fact that ASMs are also attacking the task force, nor do they inhibit the killing of ASMs due to the fact that torpedoes are also attacking the task force. Thus, these assumptions of near simultaneous attack of ASMs

and torpedoes here is strictly Blue-favorable because all ASMs and all torpedoes face the full pre-attack defenses of the task force in these models.

The second pass through the aforementioned loop in CTFMOD computes the total air-to-air attrition (as described above), it skips the ASMs versus task force calculations just described, and proceeds directly to the last portion of CTFMOD, which is to subtract the kills of aircraft from the appropriate inventories and display appropriate results. After this is done, Subroutine CTFMOD ends.

2. Major Differences Between the R-245 Model and Subroutine CTFMOD

Before discussing these differences, it should be noted that the R-245 model corresponds directly to (and only to) Subroutine CTFMOD in MEDMOD. Indeed, Subroutine CTFMOD was constructed by starting with the computer code of the R-245 model and then making changes to this code--not by building a new subroutine from the "ground up." All the other major subroutines of MEDMOD simulate resources and interactions not played in the R-245 model.[1] The major differences between the R-245 model and Subroutine CTFMOD of MEDMOD are as follows:

First, the R-245 model calculates and displays task force costs. These costing calculations and displays have been deleted from Subroutine CTFMOD.

Second, the R-245 model allows land-based AEW aircraft to be armed. As explained in Section A.3.a above, this is not allowed in Subroutine CTFMOD.

Third, the R-245 model plays only one type of Red bomber (in any one run) and does not play Red escort aircraft.

---

[1]The post-attack capability to launch power projection sorties, which is an output of the R-245 model, is calculated in Subroutine POWERP, not Subroutine CTFMOD--but this calculation is a simple multiplication based on factors determined in CTFMOD.

Subroutine CTFMOD plays Red escorts and multiple types of Red bombers.

Fourth, the R-245 model automatically determines the mix of fighter and attack aircraft on the carriers--this mix is input to MEDMOD (and to CTFMOD).

Fifth, the R-245 model only assesses attrition against Blue aircraft carriers and Red bombers, and the bomber attrition considers only the attrition that occurs before ASMs are launched. Subroutine CTFMOD assesses attrition against all Blue surface ships, against Blue fighter and attack aircraft (both in the air and on the carriers), against Red bombers (both before and after ASM launch), and against Red escort aircraft.

Sixth, the method used to determine targets for torpedoes and ASMs is more flexible in Subroutine CTFMOD than in the R-245 model.

Seventh, the R-245 model cannot automatically simulate more than one attack on the task force. MEDMOD automatically simulates multiple attacks by iterating through Subroutine CTFMOD, which updates resources and uses effectiveness parameters that can depend on the location of the task force.

H. SUBROUTINE SHPSHP

Subroutine SHPSHP models intersurface ship warfare. It comprises two interactions. First, if carriers are present, aircraft from the carriers destroy Red surface ships. If there are no Red surface ships left after this, the subroutine ends. Otherwise, a Blue surface ship versus Red surface ship battle takes place; attrition occurs on both sides.

Table 6 lists inputs used in Subroutine SHPSHP(L,ITP). The subroutine parameter L is the current location (region) of the task force, ITP is the current time period, used only

Table 6. Inputs and Major Indexing Variables Used in Subroutine SHPSHP(L,ITP)

```
INPUT PARAMETERS USED IN SUBROUTINE SHPSHP
   *BUCAP                    SSFBAK(KBA,KRSS)
   *ENACDS(KRS)              SSFRSV(KRSS,L)
    ISSBR                    SSPBDR
    ISSRB                    SSPBKR
   *PAFCNF                   SSPRDB
   *PFFCNF                   SSPRKB
    SSBACR(KRSS)             SSPRKC
    SSCFA
   *THIS VARIABLE IS ALSO USED IN ANOTHER SUBROUTINE

RESOURCE VARIABLES USED IN SUBROUTINE SHPSHP
    XPLAT(W/ XEFFCM)         XUPGS
    XEAAW                    XATTCK
    XEASWA                   XFGHTR
    XEASWN                   RS(KRS,L),KRS≥3

INDEXING VARIABLES USED IN SUBROUTINE SHPSHP
    KBA
    KRS
    KRSS
    L
```

for output purposes. Subroutine SHPSHP makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block	Variable	--	COMMON Block
ATSORU	--	COMSOR	FTSORU	--	COMSOR
ATTCKI	--	COMCTF	XCAPST	--	COMCTF
FGHTRI	--	COMCTF	XEFFCM	--	COMCTF

All Blue surface ships participate in the interaction but only an input fraction, SSFRSV(KRSS,L), of Red surface ships of kind KRSS in the region participate--the rest neither engage Blue ships nor are vulnerable to them. Two indexing systems for kind of surface Red ship are used. Red ship kinds KRS = 1 and KRS = 2 are submarines, which do not participate in this subroutine. Red ship kinds KRS = 3

through NKRS are Red surface ships. However, the notation KRSS, kind of Red surface ship, where KRSS = KRS - 2, is also used to index Red surface ships. KRSS varies from 1 to NKRSS (number of kinds of Red surface ship), where NKRSS = NKRS - 2. (The input variable NKRS must be at least 3; if it is not, the program stops.) RSSV(KRSS), which equals RS(KRSS+2,L) * SSFRSV(KRSS,L), is the number of Red surface ships of kind KRSS participating (i.e., vulnerable) in the interaction. (If no Red surface ships participate, the subroutine ends.)

If carriers are present, the numbers of Blue attack and fighter aircraft available are initially computed as

$$\text{AA} = \min\ (\text{XATTCK},\ \text{ATTCKI} * \text{XEFFCM})$$
$$\text{FA} = \min\ (\text{XFGHTR},\ \text{FGHTRI} * \text{XEFFCM})$$

These numbers are then adjusted for the need to staff CAP stations and for sorties used up earlier in the clock time period. One fighter aircraft is equivalent to SSCFA of an attack aircraft in antiship combat capability (SSCFA is input), so the weighted number of Blue aircraft available is BACA = AA + SSCFA*FA, where AA and FA are the initial values just described.

A Red surface ship of kind KRSS can be destroyed by an attack of SSBACR(KRSS) Blue aircraft (counting Blue aircraft that might be killed, by Red shipboard SAMs, for example, in the process). SSBACR(KRSS) is an input. Thus, the number of (weighted) Blue aircraft required to destroy all the Red surface ships participating is

$$\text{BACR} = \sum_{\text{KRSS}=1}^{\text{NKRSS}} \text{RSSV(KRSS)} * \text{SSBACR(KRSS)}\ .$$

If BACA $\geq$ BACR, then: (a) BACR (weighted) Blue aircraft attack the Red ships (with attack aircraft only if there are enough attack aircraft, otherwise fighters make up the shortfall),

(b) all Red surface ships participating are destroyed, and (c) the subroutine ends after assessing attrition to Blue aircraft that occurs in the course of attacking the Red ships. If BACA < BACR, then attrition to the attacking BACA (weighted) Blue aircraft is assessed, and attrition to the Red surface ships is calculated and assessed proportionately--the fraction BACA/BACR of each kind of Red surface ship is killed. The remaining Red surface ships, denoted by RSS(KRSS), go on to fight Blue surface ships.

The surface ship versus surface ship interaction allows for two different attack protocols for each side, governed by the input variables ISSBR and ISSRB. These inputs have the following meaning:

- If ISSBR = 0, different Blue ships perform detections independently of one another.
- If ISSBR = 1, the task force, as an integrated unit, detects or fails to detect each Red ship.
- If ISSRB = 0, a given Red ship detects Blue ships independently of one another.
- If ISSRB = 1, a given Red ship either detects the whole task force or detects no Blue ships at all.

The meaning of the input detection probabilities SSPBDR and SSPRDB depends on the values of ISSBR and ISSRB. In any case, each ship randomly chooses from the set of targets it has detected one target to attack and kills it with probability SSPBKR (Blue against Red), SSPRKC (Red against carriers), or SSPRKB (Red against Blue non-carriers). These probabilities are input. The rigorous statement of assumptions and derivations of the attrition equations used here will be described in a later version of this paper.

Carriers are not killed. Suppose the calculations indicate that, had the ship not been a carrier, it would have been killed with probability p. Then the variable XPLAT is left unchanged, but the carrier effectiveness XEFFCM is updated by the rule:

$$XEFFCM^{(new)} = XEFFCM^{(old)} * (1-p) .$$

Fighter and attack aircraft on the carriers, XFGHTR and XATTCK, can suffer attrition due to shots from the Red surface ships. The input ENACDS(KRS) (for KRS = 3 through NKRS) is used to compute this attrition.

Quantities are then updated and output as appropriate.

I. SUBROUTINE POWERP

Subroutine POWERP generates power projection sorties for aircraft from carriers in the task force. The targets of these power projection sorties may be defended by Red SAMs. If so, these sorties must penetrate the Red SAM defenses in order to release their ordnance against these targets. POWERP does not explicitly model the targets of these power projection sorties; it does model the Red SAMs and the Blue aircraft versus Red SAM interactions, and it accumulates successful power projection sorties flown as a measure-of-effectiveness of the task force.

Table 7 lists inputs used in Subroutine POWERP(L,ITP). The subroutine parameter L is the current location (region) of the task force, and the subroutine parameter ITP is the current time period. Subroutine POWERP makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
ATSORU	--	COMSOR
ATTCKI	--	COMCTF
CWPPAS	--	COMOUT
FGHTRI	--	COMCTF
FTSORU	--	COMSOR
PPSORT	--	COMOUT
XCAPST	--	COMCTF
XEFFCM	--	COMCTF

Table 7. Inputs and Major Indexing Variables Used in Subroutine POWERP(L,ITP)

```
INPUT PARAMETERS USED IN SUBROUTINE POWERP
   *BUCAP            PPCAL(L)          PPPKAS(KRS)
    IPPAF            PPFASM(KBA)       PPPKSA(KRS,KBA)
    IPPAW            PPFASS(KBA)       PPPSAS(KBA,KRS)
    PPAEGS(KBA)      PPFSVS(KRS)       PPSCPR(KBA,L)
    PPAVLS(KRS,L)    PPPDAS(KBA)       PPTSCS(KRS)
    PPAVSS(KRS)      PPPDSA(KRS)       WFPPAS(KBA,L)
   *THIS VARIABLE IS ALSO USED IN ANOTHER SUBROUTINE

RESOURCE VARIABLES USED IN SUBROUTINE POWERP
    PPANMS(KRS)
    PPRSAM(KRS)
    XPLAT(W/ XEFFCM)
    XATTCK
    XFGHTR

INDEXING VARIABLES USED IN SUBROUTINE POWERP
    KBA
    KRS
    L
```

The first calculation made in Subroutine POWERP is to calculate the number of sorties by type of aircraft (attack aircraft or fighter aircraft) that will be flown during the time period in question. This calculation considers the number of attack and fighter aircraft still alive on the carriers (XATTCK and XFGHTR, respectively), the relative capability of the carriers to launch sorties (XEFFCM), the number of fighter aircraft that are reserved for staffing CAP stations (BUCAP * XCAPST), the number of attack and fighter aircraft that have already "used up" all their sorties during the clock time period (ATSORU and FTSORU, respectively), and the input sortie rates used for flying power projection missions, PPSORR(I,L), where I = 1 denotes attack aircraft, I = 2 denotes fighter aircraft, and L is the location of the task force. If there are no power projection sorties flown this time period, Subroutine POWERP ends; otherwise it continues.

Next, Subroutine POWERP calls Subroutine ATRTSS to compute attrition for the Blue power projection aircraft versus Red defending SAMs interaction. (Subroutine ATRTSS is also called by Subroutine ABATCK, as described in Section M below, to compute attrition for the Blue airbase attack aircraft versus defending Red SAMs interaction.) The input fraction PPFASS(I) of attacking Blue aircraft of type I (where I is as above) attempt to suppress the Red SAMs, and so both Red SAMs and Blue aircraft can be killed in this interaction. Outputs of Subroutine ATRTSS include the numbers of SAMs (by type) that are alive and were not suppressed, that are alive but were suppressed, and that were killed, and the numbers of attacking Blue aircraft (by type) that are alive and successfully completed their mission (by delivering ordnance on ground targets other than SAMs), that are alive but were forced (by the SAMs) to jettison their ordnance, and that were killed. Details concerning Subroutine ATRTSS will be given in a later version of this documentation.

Subroutine POWERP stores the (absolute) number of successful power projection sorties flown during the time period in the variable PPSORT (where successful sorties do not count sorties factored out to do SAM suppression in support of power projection). Subroutine POWERP also accumulates (in variable CWPPAS) the total weighted number of successful power projection sorties flown so far; the weighting factor used is WFPPAS(I,L) where I denotes the type of Blue aircraft as above and L is the location of the task force at the time that the sorties are flown. (That is, just after Subroutine POWERP completes its calculations for time period ITP, the variable CWPPAS contains the total weighted number of successful power projection sorties flown summed over all time periods from 1 to ITP.) Values of the variables PPSORT and CWPPAS are displayed on the summary results table.

After updating PPSORT and CWPPAS, Subroutine POWERP prints out its results and ends.

J. SUBROUTINE ADDMOE

Table 8 lists inputs used in Subroutine ADDMOE(ITP,ISTOP).

Table 8. Inputs and Major Indexing Variables Used in Subroutine ADDMOE(ITP,ISTOP)

```
INPUT PARAMETERS USED IN SUBROUTINE ADDMOE
    (NONE)

RESOURCE VARIABLES USED IN SUBROUTINE ADDMOE
    XPLAT(W/ XEFFCM)
    XEAAW
    XEASWA
    XEASWM
    XURGS

INDEXING VARIABLES USED IN SUBROUTINE ADDMOE
    (NONE)
```

The subroutine parameter ITP gives the current time period, and ISTOP is as described in Section A.2 above. Subroutine ADDMOE makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
NTPSIM	--	COMOUT
XEFFCM	--	COMCTF

Subroutine ADDMOE is quite short and, with one exception has been thoroughly described in Section A, above. The one exception is that the values of the working variables ISTOP and NTPSIM can be changed in this subroutine. ISTOP, which is initially set to 0, is set to 1 here if the simulation is to be terminated. NTPSIM, which is initially set to 0, is set to the number of time periods simulated so far if ISTOP = 1. Since the simulation is terminated if

ISTOP = 1, NTPSIM gives the number of time periods actually simulated in a run of MEDMOD. Since ISTOP = 1 if (and when) the task force loses all effectiveness, NTPSIM can be less than the input MAXTP. (As an aside, the name for this subroutine came from the original intent to "add up measures-of-effectiveness" here, as well as to determine whether to stop the simulation. Measures-of-effectiveness, in terms of summary outputs, are now either calculated in relevant subroutines for results specific to those subroutines, or are totaled in Subroutine PRTSUM.)

K. SUBROUTINE MOVTF

Subroutine MOVTF determines (via function LOCTFF) whether or not the task force is to move during period ITP. If the task force (heading in either direction) must cross a Red-controlled submarine barrier, then MOVTF computes attrition to the ships (including direct-support submarines) in the task force, and it computes counterkills to the Red barrier submarines.

Table 9 lists inputs used in Subroutine MOVTF(LOCTF,ITP). LOCTF is the location (region) of the task force at the beginning of time period ITP. Subroutine MOVTF makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
SCK32		BARSCK
XEFFCM		COMCTF

Subroutine MOVTF calls function LOCTFF ("locate task force function") to determine the desired location of the task force at the beginning of period ITP + 1. This location is determined by two input vectors, LGTHMP(I) (length of movement period I), and LTFMP(I) (location of the task force during movement period I), where I ranges from 1 to the input limit MIMP, and indexes "movement period." The task force

Table 9. Inputs and Major Indexing Variables Used in Subroutine MOVTF(LOCTF,ITP)

```
INPUT PARAMETERS USED IN SUBROUTINE MOVTF
     ATTWGT                   CSCDWO
     BACCDW(KBS)              FM3(KBS)
     BACPCK(KBS)             *ICTL(IBAR)
    *BARLTH(IBAR)            +LGTHMP(I)
     BECDW(KBS)              +LTFMP(I)
     CACDWO                  +MIMP
     CPAGV                    REDW(KBS)
     CPBPK(KBS)               TPAS
     CPRSCK(KBS)              WTFCBO
    *THIS VARIABLE IS ALSO USED IN SUBROUTINE MOVRS
    +THIS VARIABLE IS USED IN FUNCTION LOCTFF

RESOURCE VARIABLES USED IN SUBROUTINE MOVTF
     BSSNDS                   XEASWN
     XPLAT(W/ XEFFCM)         XURGS
     XEAAW                    RSIBAR(IBAR)
     XEASWA

INDEXING VARIABLES USED IN SUBROUTINE MOVTF
     I
     IBAR
     KBS
```

remains in the region LTFMP(I) throughout movement period I. Thus, for example, LGTHMP(1) = 2 and LGTHMP(2) = 3 means that movement period 1 encompasses time periods 1 and 2, and movement period 2 encompasses time periods 3, 4, and 5. For the last time period in movement period I, the task force is in Region LTFMP(I) at the <u>beginning</u> of that time period, but is in Region LTFMP(I+1) at the <u>end</u> of the time period. (That is, MOVTF is called in the middle of the loop over time periods; therefore, if the task force is to change regions, it does so <u>during</u> a time period.)

Subroutine MOVTF determines the destination DESTF of the task force. (The working variable DESTF is declared integer.) Two STOP statements have been put into this subroutine as checks; they occur if DESTF > NLOC or if

$|DESTF - LOCTF| \geq 2$. (That is, the task force cannot move more than one region away from its current location during a single time period.) See Table B-10 of Appendix B for more information on these STOPs. These stops can be avoided by manipulating the input array LTFMP.

If DESTF equals LOCTF (i.e., the task force does not move) or there is no barrier or a Blue-controlled barrier between regions LOCTF and DESTF, no updating of the task force ships is necessary. LOCTF is updated appropriately and the subroutine ends. If there is a Red-controlled barrier, attrition must be assessed.

There are three possible protocols that can be used when crossing a Red-controlled barrier. If carriers are present, only Protocol 3 is used; it is discussed below. Protocol 2 is used if there are Blue ships present with some air ASW capability. It is a two-step procedure identical to the one described in Subroutine MOVRS, with Red and Blue reversed. Protocol 1, used if no Blue ships have air ASW capability, is also identical, mutatis mutandis, to the one in MOVRS.

Protocol 3 is a three-step crossing procedure. Instead of crossing along the whole barrier length BARLTH(IBAR), the task force crosses along a front of width

$$WTFCB = \min(WTFCB0, BARLTH(IBAR)) ,$$

where variable WTFCB0 is an input. (Of course, BARLTH(IBAR), the physical length of the barrier between Regions IBAR-1 and IBAR, is also an input.) Since the barrier submarines are assumed to be evenly spaced along the total length of the barrier, this results in the task force being vulnerable to fewer barrier submarines.

The first step of the crossing procedure is similar to the first step of Protocol 2: ASW aircraft from the carrier

and escort ships, as appropriate, attempt to kill barrier submarines. Then the Blue ships transit. The input fraction FM3(KBS) of the ships of kind KBS transit on Step 3; 1-FM3(KBS) transit on Step 2. See Table B-6 of Appendix B for a precise description of variable KBS. All carriers move on Step 3. (The point here is that barrier submarines counterkilled in Step 2 cannot attack Blue ships transiting in Step 3.) In both Steps 2 and 3, barrier submarine counterkills and attrition to non-carrier Blue ships is assessed by Subroutine BARKCK; the procedure is identical, mutatis mutandis, to the one described above in Subroutine MOVRS. In Step 3, Red barrier submarines attacking carriers can degrade the average relative capability, XEFFCM, of the carriers.

After all the attrition to Blue ships and Red barrier submarines has been computed, results are output and variables (including LOCTF) are updated as appropriate.

L. SUBROUTINE MOVRS

Subroutine MOVRS moves Red ships (including submarines) from region to region. In transit, some Red ships might have to cross some submarine barriers controlled by Blue. If this is so, attrition to Red ships, by type, and counterkills of the Blue barrier submarines are computed.

Table 10 lists inputs used in Subroutine MOVRS(LOCTF,ITP). (This subroutine does not use any labeled COMMON blocks.) The point of departure is the matrix RS(KRS,LOC) which represents the number of Red ships of kind (type) KRS (KRS = 1, ..., KNRS) in Region LOC (LOC = 1, ..., NLOC). There are in fact NLOC + 1 regions, numbered 0 through NLOC, but Region 0 is a sanctuary for Blue and thus never contains Red ships.)

The current region, LOCTF, of the task force is input via the parameter list. Subroutine MOVTF has already been

Table 10. Inputs and Major Indexing Variables Used in Subroutine MOVRS(LOCTF,ITP)

```
INPUT PARAMETERS USED IN SUBROUTINE MOVRS
     AWRCBB                  *ICTL(IBAR)
    *BARLTH(IBAR)             PRSM(KRS,LOC,LOCTF1)
     BEDW(KRS)                RACCDW(KRS)
     CPBSCK(KRS)              RACPCK(KRS)
     CPRPK(KRS)               RECDW(KRS)
    *THIS VARIABLE IS ALSO USED IN SUBROUTINE MOVTF

RESOURCE VARIABLES USED IN SUBROUTINE MOVRS
     BSIBAR(IBAR)
     RS(1,LOC)
     RS(2,LOC)
     RS(KRS,LOC),KRS≥3

INDEXING VARIABLES USED IN SUBROUTINE MOVRS
     IBAR
     KRS
     LOC
     LOCTF1
```

called in the current time period, and LOCTF is the most recent region of the task force. (Thus, it is assumed that Red knows this region.) Since LOCTF can vary from 0 through NLOC, the variable LOCTF1 = LOCTF + 1 is also used here to index the task force region (so that the index LOCTF1 is always strictly positive).

The protocol for moving Red ships is as follows. If the task force is in Region LOCTF, then of the RS(KRS,LOC) Red ships of kind KRS in Region LOC, the input fraction PRSM(KRS,LOC,LOCTF1) move one region toward the task force--PRSM stands for "proportion of Red ships moving." More technically, ships in Region LOC move into Region NEWLOC, where NEWLOC is given by

$$\text{NEWLOC} = \text{LOC} + \frac{\text{LOCTF} - \text{LOC}}{|\text{LOCTF} - \text{LOC}|}\ , \quad \text{for LOC} \neq \text{LOCTF}\ .$$

Of course, Red ships in Region LOCTF do not move, and no Red ships move into Region 0.

If there is no submarine barrier between Regions LOC and NEWLOC, or the barrier is Red-controlled, there is no attrition. For Blue-controlled barriers, attrition is computed by Subroutine BARKCK, which accepts as inputs the number of ships attempting transit by kind, the number of barrier submarines, and several effectiveness parameters, and outputs the expected numbers of transiting ships killed, by kind, and barrier submarines counterkilled. (The assumptions underlying the modeling of combat in BARKCK will be described thoroughly in a later version of this documentation.)

If the Red ships have any air ASW capability, a two-step process is used. First ASW aircraft from the ships kill those barrier submarines they can (these aircraft are not explicitly modeled; their effectiveness is represented through inputs). Then the Red ships (including submarines) transit the barrier, and are subject to attrition by the remaining barrier submarines. (These submarines can be counterkilled by the Red ships.) Of course, if Red has no air ASW capability, only the second step takes place. Red air ASW capability is determined by whether the variable

$$SP = \sum_{NRS=1}^{NKRS} RACCDW(KRS) * RACPCK(KRS)$$

is greater than zero or not.

The barrier crossing procedure is performed for every Blue-controlled barrier. The matrix RS(KRS,LOC) is updated to reflect movement and barrier kills and the results are displayed in the detailed output for the time period in question.

There can (optionally) be both a Blue submarine barrier and a Red submarine barrier between two adjacent regions; however, in this case, no direct interactions between the barriers are simulated.

M. SUBROUTINE ABATCK

Subroutine ABATCK models attack of the vulnerable Red airbase by Blue fighter and attack aircraft from the carriers. The Red airbase is defended by Red aircraft and SAMs; ABATCK calculates the attrition to Blue aircraft caused by these Red defenses, attrition to the defenses, and finally, Red aircraft killed on the airbase.

Table 11 lists inputs used in Subroutine ABATCK(L).

Table 11. Inputs and Major Indexing Variables Used in Subroutine ABATCK(L)

```
INPUT PARAMETERS USED IN SUBROUTINE ABATCK
     AAAEDA(KRD)          ABESGS(KRA)           IAAED
     AAAEDE(KRD)          ABFASS(KBA)           IABAEQ
     AAAEED(KBE)          ABFSM(KBA)            IABAF
     AACA                 ABFVS(KRSAM)          IABAW
     AAPAJO(KBA)          ABPDA(KBA)            IKRAS(KRA)
     AAPDDA(KRD)          ABPDS(KRSAM)         *PAFCNF
     AAPDDE(KRD)          ABPKA(KRSAM)          PARK
     AAPDED(KBE)          ABPKS(KRSAM,KBA)      PASS(I)
     AAPKAD(KBA,KRD)      ABPSA(KBA,KRSAM)      PBDRN(I)
     AAPKDA(KRD,KBA)      ABTSC(KRSAM)          PBDRS(I)
     AAPKDE(KRD,KBE)      ABVGSS(KRSAM)         PBKRN(I)
     AAPKED(KBE,KRD)      AVALED(L,IATF)        PBKRS(I)
     AASRAA(L)           *BUCAP                *PFFCNF
     AASRED               FAACA(L)              RARBAR(K)
     AASRFA(L)            FACOB(KRA,IATF)       XIA(L)
     AASRFE(L)            FFACA(L)              XIE(L)
     AASRID               FFACE(L)              XNRAB
     ABAVLS(KRSAM)        FHSK(I)
     ABCAS                IAADA
    *THIS VARIABLE IS ALSO USED IN ANOTHER SUBROUTINE

RESOURCE VARIABLES USED IN SUBROUTINE ABATCK
     XPLAT(W/ XEFFCM)     ATABT(IAB,KRB)        SHEL
     XATTCK               AESCAB(IAB)           ABANM(KRSAM)
     XFGHTR               AINTCT                ABRSAM(KRSAM)

INDEXING VARIABLES USED IN SUBROUTINE ABATCK
     I                    KBA                   KRD
     IAB                  KBE                   KRSAM
     IATF                 KRA                   L
     K                    KRB
```

Subroutine ABATCK makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
ATSORU	--	COMSOR
ATTCKI	--	COMCTF
FGHTRI	--	COMCTF
FTSORU	--	COMSOR
NTPSLA	--	COMGA
XCAPST	--	COMCTF
XEFFCM	--	COMCTF

If there are insufficient Red aircraft to warrant an attack by Blue (this is determined by input variable RARBAB) the subroutine ends. Otherwise, the numbers of attack and escort Blue aircraft from the carrier are determined, taking into account (input) allocations and the number of fighter aircraft needed to staff CAP stations. The aircraft are multiplied by the appropriate sortie rate (see Table 12) to yield the numbers of sorties flying. Sortie rates are generally considered as being less than 1.0; the interpretation is that the number of sorties is the number of aircraft actually flying. (However, the model can handle sortie rates greater than 1.0.) If there are insufficient Blue attack or escort sorties to perform airbase attack (the requirements are given by the inputs XIA(L) and XIE(L), respectively) the subroutine ends. Otherwise, Subroutine AIRAIR is called, which simulates and computes attritions for the following two interactions, in this order:

1. Blue fighter aircraft on escort missions versus Red fighter and interceptor aircraft on defense missions.
2. Blue attack and fighter aircraft on attack missions versus surviving Red fighter and interceptor aircraft on defense missions.

Table 12. Aircraft/Mission Combinations Modeled in ABATCK

	Aircraft Variable(s)	Sortie Variable
Blue attack aircraft performing airbase attack	AA	SA(1)=AA*AASRAA(L)
Blue fighter aircraft performing airbase attack	FA	SA(2)=FA*AASRFA(L)
Blue fighter aircraft performing airbase attack escort	FE	SE(1)=FE*AASRFE(L)
Red bombers, by kind (KRB=1, ..., NKRB)	A(1) through A(NKRB)	(n/a)
Red escort aircraft (total)	A(NKRB+1)	(n/a)
Red escort aircraft performing defense	ED	SD(1)=ED*AASRED
Red interceptor aircraft (all perform defense)	A(NKRA) and ID[a] (NKRA=NKRB+2)	SD(2)=ID*AASRID

[a]Variable ID is declared real in the program.

Most of the input parameters starting with "AA" are detection or kill probabilities or other parameters for these interactions.

For each aircraft/mission combination, one result is a certain number of sorties killed. For Blue aircraft on attack missions, a calculated number are engaged and killed, a calculated number are engaged, are not killed, but jettison their ordnance and return to the carriers, and the rest continue on their attack mission. Blue escort aircraft and Red defense aircraft complete their missions in these interactions and so all sorties on these missions that are not killed fly home after these interactions.

Next, the Blue attack/Red SAM interaction is simulated by calling Subroutine ATRTSS. NABSAM kinds of Red SAMs are played; the current dimensioning of MEDMOD restricts NABSAM to be either 1 or 2. Results of ATRTSS are: numbers of SAM fire control centers killed, by kind, actual missiles expended, by kind, and Blue attack and fighter aircraft killed, flying home, and continuing on to attack the Red airbase. Every Blue attack sortie that continues to attack

aircraft and shelters on the airbase makes an input number, given by PASS(I), of equal-effectiveness passes at these targets (I = 1 is for attack aircraft, I = 2 is for fighter aircraft performing the attack mission).

There are multiple kinds of Red aircraft on the ground, indexed by kind KRA. KRA = 1 through NKRB denotes kinds of Red bombers, KRA = NKRB + 1 denotes Red fighter aircraft, and KRA = NKRB + 2 denotes Red interceptors. There is an input number, SHEL, of aircraft shelters. A priority sheltering scheme is used: aircraft of kind KRA = 1 are sheltered first, if there are any shelters left, aircraft of kind KRA = 2 are sheltered, and so forth. This is subject to the provision that a particular kind of aircraft can fit into shelters. The input IKRAS(KRA) ("is this kind of Red aircraft shelterable?") determines this: if IKRAS(KRA) = 1, kind KRA aircraft are shelterable; if IKRAS(KRA) = 0, they are not.

There are ATABT(1,KRB) Red bombers, AESCAB(1) Red fighters, and AINTCT interceptors stationed on the notional vulnerable Red airbase. However, only the input fraction FACOB(KRA,IATF) of Red aircraft of kind KRA are actually on the airbase when the attack occurs; the other aircraft are out flying missions. The fraction of aircraft on base can depend on whether a Red attack on the task force is planned later on in the clock time period (day) or not (IATF = 1 or 2, respectively). Even though only A(KRA) * FACOB(KRA,IATF) aircraft of kind KRA are on base, the full amount of A(KRA) aircraft are considered as shelterable (if IKRAS(KRA) = 1), where A(KRA) is ATABT(1,KRB), AESCAB(1), or AINTCT. This consideration can result in a "shell game" involving empty shelters. If IKRAS(KRA) = 0 for some KRA, then some Red aircraft cannot be sheltered and so all of these aircraft that are on the ground when the Blue attack occurs will be

in the open, even though some shelters may be empty. Also, if FACOB(KRA,IATF) < 1 for some values of KRA and IATF, then some Red aircraft may be in the open even though some shelters are empty during the Blue attack. This latter event occurs because MEDMOD assumes that no aircraft can use another aircraft's shelter while that other aircraft is out flying a mission. Thus, if FACOB(KRA,IATF) < 1, a Red aircraft of type KRA may be assigned a shelter but may be out flying a mission when the Blue attack occurs, and so its shelter would be empty at that time. If there were not enough shelters for each shelterable Red aircraft to have its own shelter, then this would mean that some shelterable Red aircraft would be in the open even though some shelters (which are assigned to aircraft that are out flying missions) are empty.

The attrition to Red from airbase attack is determined by Subroutine ATRTAB. The Red aircraft on the (notional) vulnerable airbase are assumed to be distributed over an input number, XNRAB, of identical typical (real) airbases; XNRAB should be chosen such that the size of the typical Red airbase matches the actual real airbases in the group comprising the notional vulnerable Red airbase. On each typical base there are an input number (PARK) of parking areas for nonsheltered aircraft (or for all aircraft, if IABAEQ = 3). The variable PARK should not be zero. There are three choices of attrition protocol, indexed by IABAEQ = 1, 2, or 3. These are described in detail in IDA Paper P-1111, [10]. Also see the ATRTAB code and the definition of input variable IABAEQ. The different kinds of Red aircraft have the same vulnerability to being killed on the ground, the only factor is whether they are sheltered or not. (The detection and kill probabilities are PBDRS(I), PBDRN(I), PBKRS(I), and PBKRN(I), respectively, where I denotes the type of attacking Blue aircraft, as described above.) A shelter that is hit (in such a way that an aircraft

inside it would be destroyed) is itself destroyed with probability FHSK(I).

Subroutine ATRTAB computes the number of open aircraft, sheltered aircraft, and shelters killed. ABATCK then determines Red aircraft killed by kind and updates and outputs quantities as appropriate.

N. SUBROUTINE PRTSUM

Subroutine PRTSUM totals some summary results and writes these and other results (described below) onto an output file (TAPE10) for eventual printing on the summary results table.

Table 13 lists inputs used in Subroutine PRTSUM(LC,ITP).

Table 13. Inputs and Major Indexing Variables Used in Subroutine PRTSUM(LC,ITP)

```
INPUT PARAMETERS USED IN SUBROUTINE PRTSUM
    WFTFL(L)

RESOURCE VARIABLES USED IN SUBROUTINE PRTSUM
    BSSNDS              XURGS               RSIBAR(IBAR)
    BSIBAR(IBAR)        XATTCK              RS(KRS,L),KRS.GE.3
    XPLAT(W/ XEFFCM)    XFGHTR              ATABT(IAB,KRB)
    XEAAW               PLBLBD(KBD,L)       AESCAB(IAB)
    XEASWA              RS(1,L)             AINTCT
    XEASWN              RS(2,L)

INDEXING VARIABLES USED IN SUBROUTINE PRTSUM
    IAB                 KRB
    IBAR                KRS
    KBD                 L
```

The calling argument LC denotes the current location of the task force in this subroutine, and ITP denotes the current time period. Subroutine PRTSUM makes use of the following variables stored in labeled COMMON blocks:

Variable	--	COMMON Block
CWPPAS	--	COMOUT
CWTPTF	--	COMOUT
PPSORT	--	COMOUT
XEFFCM	--	COMCTF

Subroutine PRTSUM calculates the following values for the summary results table. First it calculates the current number of Blue surface ships, excluding aircraft carriers. That is, it calculates TBSHPC, where

$$TBSHPC = XEAAW + XEASWA + XEASWN + XURGS .$$

The reason for not including carriers here is that the number of carriers, XPLAT, does not vary in MEDMOD, and its (constant) value is printed at the top of the summary results table.

Next, PRTSUM calculates the cumulative weighted number of time periods that the task force has accumulated through the current time period--this value is denoted by CWTPTF. CWTPTF is initially set to zero in Overlay MEDMOD, and it is incremented here in one of two ways, depending on whether carriers are ($XPLAT > 0$) or are not ($XPLAT = 0$) part of the task force. If carriers are part of the task force, then the old value for CWTPTF is updated in PRTSUM by adding to it the term:

$$XEFFCM * WFTFL(LC) .$$

Thus, at the end of time period ITP, CWTPTF will have the value

$$CWTPTF = \sum_{i=1}^{ITP} XEFFCM(i) * WFTFL(LC(i))$$

where XEFFCM(i) is the relative average effectiveness of the aircraft carriers in the task force at the end of time period i, LC(i) is the location of the task force at the

end of time period i, and WFTFL(LC) is an input weighting factor on these locations. If there are no carriers in the task force, then CWTPTF is updated in PRTSUM by adding to it the term:

$$TBSHPC * WFTFL(LC) .$$

In this case, at the end of time period ITP, CWTPTF will have the value

$$CWTPTF = \sum_{i=1}^{ITP} TBSHPC(i) * WFTFL(LC(i))$$

where TBSHPC(i) is the total number of Blue ships in the task force (since there are no carriers) at the end of time period i, and LC(i) and WFTFL(LC) are as described above. (If there are no carriers in the task force, then XEFFCM stays constant at 1.0 through a run of MEDMOD.)

Next, PRTSUM calculates the total number of Blue land-based interceptors alive at the end of the time period, TBLBDS, by

$$TBLBDS = \sum_{KBD=1}^{NKBDPL} \sum_{L=1}^{NLOC} PLBLBD(KBD,L) .$$

Then, PRTSUM calculates the total number of Blue submarines alive at the end of the time period, TBSUBS, by

$$TBSUBS = BSSNDS + \sum_{L=1}^{NLOC} BSIBAR(L) .$$

Then, PRTSUM calculates the total number of Red surface ships, submarines, bombers, and escorts (TRSHIP, TRSUBS, TRBMRS, and TRESCS, respectively) by

$$TRSHIP = \sum_{KRSS=3}^{NKRS} \sum_{L=1}^{NLOC} RS(KRSS,L) ,$$

$$TRSUBS = \sum_{L=1}^{NLOC} (RS(1,L) + RS(2,L) + RSIBAR(L)) ,$$

$$TRBMRS = \sum_{IAB=1}^{2} \sum_{KRB=1}^{NKRB} ATABT(IAB,KRB), \text{ and}$$

$$TRESCS = \sum_{IAB=1}^{2} AESCAB(IAB) .$$

Finally, Subroutine PRTSUM writes selected quantities onto the output file (TAPE10) for eventual printing on the summary output table. The first row in Figure 3 gives the column headings of the summary output table that are written by Overlay MEDMOD. The second row of that figure gives the names of the variables whose values are written in each column by PRTSUM. A sample (computer produced) summary results table is given at the end of Appendix D.

After writing these quantities onto the output table, PRTSUM ends.

It should be noted that, if there are no carriers in the task force, then the only useful Blue results described on the summary results table are CWTPTF (the cumulative weighted effectiveness of the task force), TBSHPC (the total number of ships in the task force), and TBLBDS (the total

Header: -- SUMMARY RESULTS FOR BLUE --

Column Heading:	ITP	XEFFCM	CUMLTV WGHTED EFCTVNS	TOTAL SURFACE SHIPS	TOTAL SUBS	FIGHTER AIRCRAFT (ON CV)	ATTACK AIRCRAFT (ON CV)	LAND-BSD INTCPTR AIRCRAFT	POWER PROJECTN SORTIES	CUMLTIVE WGHTD PP SORTIES
Variable Name:	ITP	XEFFCM	CWTPTF	TBSHPC	TBSUBS	XFGHTR	XATTCK	TBLBDS	PPSORT	CWPPAS

Header: -------------- SUMMARY OF RESULTS FOR RED -------------

Column Heading:	TOTAL SURFACE SHIPS	TOTAL SUBS	TOTAL BOMBERS	FIGHTER AIRCRAFT	INTCPTR AIRCRAFT
Variable Name:	TRSHIP	TRSUBS	TRBMRS	TRESCS	AINTCT

Figure 3. Variables Whose Values Are Tabulated on the Summary Results Table

number of Blue land-based interceptors). All other Blue results are carrier related.[1]

O. SUMMARY

Sections B through N above (supplemented by Appendix A for TIMET and by Appendices B and C for the other subroutines) are intended to give a reasonably thorough description of what each major subroutine of MEDMOD is supposed to do and a general description of how each of these subroutines works. Indeed, four subroutines (TIMET, GNAATK, ADDMOE, and PRTSUM) are fully documented here, and two others (DDAY and SUBSUB) require only that the reader refer to the comments in the code for Subroutine BINOAT and to [6] for appropriate background (see also Section 3 of [11] which summarizes the relevant results of [6] in a somewhat more comprehensive setting). The main overlay, DRIVER, and Overlays INP and MEDMOD are relatively fully documented in Section A and the appendices. Thus, only full documentation of the seven subroutines PLBAB, CTFMOD, SHPSHP, POWERP, MOVTF, MOVRS, and ABATCK (and of the subroutines they call) remains to be written.

---

[1]It should also be noted that MEDMOD does not automatically zero out the number of aircraft on carriers (XFGHTR, XATTCK, XAEW, and XASW) when there are no carriers (i.e., when XPLAT = 0.0). Thus, if a user of MEDMOD chooses to input no carriers and to input positive numbers of carrier-based aircraft, then these aircraft will be counted in MEDMOD as if they were flying missions from invisible and invulnerable ships. It is possible that such an output might be useful in rare special cases. When it is not useful, a user of MEDMOD should simply input no carrier-based aircraft in those runs in which there are no carriers.

Chapter III

LIMITATIONS

The first section below discusses some limitations of the scope of MEDMOD. These limitations are presented in a logical order, not an order of importance. The next three sections discuss limitations of MEDMOD given its scope. The most significant limitations are presented in the first of these three sections, intermediate limitations are presented in the second of these sections, and relatively minor limitations are given last. With one exception, the limitations within each of these last three sections are listed roughly in order of their importance. The one exception is that the relatively minor limitations in the last section are divided into four functional subsections. The first of these subsections contains limitations primarily concerned with geography, the second contains limitations primarily concerned with the types of resources being modeled, the third contains limitations primarily concerned with interactions between resources, and the fourth states an output-related limitation.

All limitations in scope apply both to MEDMOD and to the forthcoming improved version of MEDMOD, called NAVMOD. Limitations given scope that apply to MEDMOD, but not to NAVMOD, will be parenthetically noted. One characteristic not listed below is that, at the time this documentation was written, neither MEDMOD nor NAVMOD had been used in a major study or analysis.

A. LIMITATIONS IN SCOPE

1. MEDMOD does not simulate ground versus ground combat (i.e., Blue ground forces versus Red ground forces). Thus, for example, the timeliness and impact of power projection sorties cannot be measured in MEDMOD, and the outcome of a whole air/ground/sea war cannot be used as a measure-of-effectiveness.

2. MEDMOD does not simulate Blue land-based air attacks on Red's (land) airbases nor does it simulate Red land-based air attacks on Blue's (land) airbases. (All combat interactions simulated in MEDMOD involve, in part, either Blue ships, Blue submarines, Blue carrier-based aircraft, Red ships, Red submarines, or Red aircraft on the way to attack Blue ships.)

3. Neither Blue nor Red land-based close-air support or air interdiction is simulated. Thus, for example, MEDMOD cannot address questions like: "Would Blue be better off if it had fewer carrier-based aircraft to fly power projection missions and had more land-based aircraft to fly close-air support or interdiction missions instead, or would Red be better off if it used its aircraft to attack Blue ground forces instead of attacking the Blue task force?"

Note that MEDMOD allows Blue land-based aircraft to provide defense for the task force, but (as stated in limitations 2 and 3 above) it does not allow land-based aircraft to perform the other missions of carrier-based aircraft (such as power projection or attack of Red airbases). These two limitations might not be significant if land-based aircraft can defend the task force only when the task force is in areas in which its aircraft cannot perform either power projection or airbase attack missions. However, if there are areas in which land bases can provide significant air defense for the task force and, at the same time, aircraft from the task

force can fly power projection missions or can attack Red airbases (or both), then these limitations could be quite severe because, in this case, an equal cost force of land-based aircraft might perform these missions much more effectively than the carrier-based aircraft. (Of course, if MEDMOD were expanded to include Blue land-based aircraft flying power projection and airbase attack missions, then Red aircraft should also be able to attack Blue land bases and, perhaps, to fly power projection missions to mitigate the effect of the power projection missions flown by Blue.)

4. Protection of sea lines of communication is not directly simulated in MEDMOD.

5. Chemical warfare is not simulated in MEDMOD. (Nuclear warfare can be simulated in a manner consistent with MEDMOD's level of aggregation by using suitable inputs.)

B. MAJOR LIMITATIONS WITHIN MEDMOD'S SCOPE

1. MEDMOD is a relatively highly aggregated model (like the R-245 model [1] or the model discussed in [2], as compared to detailed models like IDACASE [5], FLOATS [12], or NADS [13]). The fact that MEDMOD is relatively highly aggregated is a characteristic, not a limitation, of MEDMOD, but this characteristic means that MEDMOD has the advantages and limitations pertaining to aggregated models. One advantage is that it simulates many different interactions relatively quickly (in terms of computer running time), and so it can be used both as an integrating model to examine the impact of combining the effects of many individual interactions, and as a parametric model to analyze many different cases. The corresponding limitation is that MEDMOD does not simulate any particular interaction in great detail--each individual simulation is relatively simple. Whether these individual simulations are overly simplistic or not depends on the

simulation, the inputs, and the intended use of MEDMOD. Each of these individual simulations could be expanded in detail, but not all such expansions necessarily lead to a net improvement. Rather than display a list of various possible expansions for each individual simulation in MEDMOD as a list of potential limitations of the current program, we simply note that MEDMOD is a highly aggregated model.

2. In general, ordnance supply and consumption is not simulated in MEDMOD. The reason that this limitation is relatively important is threefold. First, shortages (even local shortages) of ordnance on either side can be quite significant. Second, shortages (especially local shortages) may be likely. Third, ordnance supply and consumption could reasonably be modeled at MEDMOD's (high) level of aggregation. Until this limitation is removed, mid-run inputs using Subroutine TIMET can be used to reduce the effectiveness or number of resources due to the consumption of ordnance (and, perhaps, of other supplies).

C. INTERMEDIATE LIMITATIONS WITHIN MEDMOD'S SCOPE

1. Ship-based and submarine-based cruise missile attacks on the other side's ground forces and airbases are not simulated (for either Red or Blue) in MEDMOD. (NAVMOD will simulate Blue ship-based and submarine-based cruise missiles used for power projection, but not airbase attack.)

2. Attrition of Blue AEW aircraft is not simulated, and the sortie rate of carrier-based AEW aircraft is unaffected by damage to the carrier. (NAVMOD will simulate attrition to Blue AEW aircraft while on the carrier, and it will degrade their sortie rates due to carrier damage. Land-based AEW aircraft are invulnerable to attack both in MEDMOD and in NAVMOD.)

3. Attrition of the jamming capability on either side is not simulated in MEDMOD. Indeed, neither Red nor Blue

jammers are explicitly simulated (though their impact can be implicitly simulated by suitably adjusting appropriate inputs).

4. Blue land-based air attack of Red surface ships is not simulated in MEDMOD. (This limitation is removed in NAVMOD.)

5. Mines are not simulated in MEDMOD.

6. Red battlefield defense aircraft (that could protect Red ground forces against Blue power projection missions) are not simulated in MEDMOD. The reason that this limitation is not extremely significant is that, in cases where the impact of Red battlefield defense aircraft could be important, their effectiveness could be roughly represented by adding a "dummy" type of Red SAM, whose effectiveness characteristics would correspond (as closely as possible) to Red battlefield defense aircraft, not to a real type of Red SAM.

7. Losses of URG ships do not degrade task force capability in MEDMOD. (This limitation is partially removed in NAVMOD.)

D. RELATIVELY MINOR LIMITATIONS WITHIN MEDMOD'S SCOPE

1. Limitations Concerning Geography

1. All Blue ships must be in the same Blue task force, and so all Blue ships (and non-barrier submarines) must be in the same geographical region at any point in time. The reason that this limitation is not of higher importance is as follows. Suppose there were two sets of Blue ships in two different regions, and that (at least) one of these sets of ships contained no aircraft carriers. Then MEDMOD could be run twice, once with (only) one set of Blue ships and the Red resources assigned (or likely) to attack that set of ships, and once with (only) the other set of Blue ships and

with the Red resources assigned (or likely) to attack that other set of ships (the two sets of Red resources should be mutually exclusive here). If the two sets of Blue ships were to combine to form one task force (or if Red resources were reallocated) at some time during the simulation, then this could be accomplished in MEDMOD by adjusting appropriate resource values at the appropriate time using Subroutine TIMET. This combining separate runs approach would also be useful if each of the two sets of Blue ships contained aircraft carriers but, for reasons of capability, strategy, or scenario, the aircraft from (at least) one of these sets of carriers are not used to attack Red airbases. If a scenario called for two carrier task forces in two different regions and aircraft from both forces are to attack Red airbases, then this limitation could be significant for that scenario.

2. No combat between ships (or ship-based resources) that are in different geographical regions is allowed in MEDMOD. For example, Blue carrier-based aircraft cannot attack Red surface ships that are not in the same geographical region as the task force.

3. All Red SAMs that are defending against Blue power projection are located in one notional area; they are not directly associated with geographical regions.

4. The geographical regions must be numbered sequentially, and ships and submarines cannot move more than one region per time period. This restriction would probably be relatively easy to relax for any particular situation, but it would be tedious (and may be computer storage space consuming) to automatically allow all mathematically possible combinations of moves.

5. A problem can occur if the task force moves into Region 0 (from Region 1) and then moves out of Region 0 (back into Region 1). Specifically, Red bombers must "rest" for

IATKRT(L) - 1 time periods between attacks of the task force (given that the task force is in Region L), where IATKRT(L) is input. As currently programmed, MEDMOD does not count time periods that the task force spends in Region 0 in determining when Red bombers can next reattack the task force. This limitation would be quite easy to remove if it were desirable to model a case in which the task force moves in and out of Region 0, but since this limitation only applies when the task force moves back and forth this way, it may never be necessary to remove this limitation.

2. Limitations Concerning Resources

1. MEDMOD can simultaneously play multiple types of some resources, but it can only play one type of the following Blue resources (in any one run): aircraft carriers, AAW escorts, air-capable ASW escorts, non-air capable ASW escorts, URG ships, submarines in direct support of the task force, submarines in barriers, carrier-based attack aircraft, carrier-based fighter aircraft, carrier-based AEW aircraft, carrier-based ASW aircraft, land-based AEW aircraft, and land-based ASW aircraft. Also, MEDMOD can only play one type of the following Red resources (in any one run): non-barrier torpedo-firing submarines, non-barrier ASM-firing submarines, submarines in barriers, fighter aircraft, interceptor aircraft, and shelters for aircraft. The reason that this limitation is not extremely significant (recognizing the fact that, in reality, there are multiple types of all of these resources) is related to the level of aggregation of MEDMOD. With its relatively high level of aggregation, MEDMOD often computes the total capability of a group of resources as the number of resources in this group times the capability of each resource. For example, the total AAW capability of the AAW escorts is computed as the number of AAW escorts times the average AAW capability of each AAW escort. Thus, MEDMOD would

assess the same total effectiveness to a defense consisting of three high quality AAW escorts and three low quality AAW escorts as it would to a defense consisting of six average notional AAW escorts (whose individual quality is an equal average of the high and low quality escorts). Therefore, assuming the vulnerabilities of these escorts are about equal, the level of aggregation of MEDMOD is such that multiple types of AAW escorts of differing quality (and, in general, multiple types of other resources) can be appropriately simulated by using notional types of resources of average quality.

With one exception, this limitation concerning multiple types of resources also applies to NAVMOD. The exception is that NAVMOD will allow all Blue surface ships to be potentially air-capable (e.g., all Blue surface ships can be capable of launching LAMPS sorties, though with different sortie rates depending on the type of Blue ship). Thus, in NAVMOD, there will be no fundamental distinction between air-capable ASW escorts and non-air capable ASW escorts. Accordingly, a user of NAVMOD could aggregate all ASW escorts into one notional type of ASW ship, which would "free up" one type of Blue ship for other use. For example, all ASW escorts could be lumped into one average type of ASW escort, and the variables used for the other type of ASW escort could be used to simulate battleships or any other type of Blue surface ship not currently simulated in MEDMOD or NAVMOD. (Some minor reprogramming of NAVMOD would be required to allow the different vulnerability of battleships and ASW escorts to be represented, but this reprogramming would be easy to do.)

2. Shore (ground)-to-ship missiles are not simulated in MEDMOD for either side. This limitation includes both ballistic missiles and cruise missiles, and it includes both conventionally tipped and nuclear tipped missiles. A related but relatively less important limitation is that long-range

ship-to-ship (and submarine-to-ship) missiles are not simulated in MEDMOD. In particular, this means that: (a) ships in one region cannot shoot at ships in other regions (as was stated in Limitation 2 of Section D.1 above), and (b) Red surface ships are vulnerable to attack by Blue carrier-based aircraft before the Red ship can fire missiles at the Blue task force and before the Blue task force can fire its ship-based missiles at the Red attacking ship.

3. Red submarines in regions either are either missile-firing submarines or torpedo-firing submarines, but not both. Again, this limitation is relatively not important because, for example, two Red submarines that can fire half a load of missiles, then move in to fire half a load of torpedoes, can be modeled as one torpedo-only submarine and one missile-only submarine. This approach for modeling Red submarines that would be used to fire both torpedoes and missiles is not rigorously correct, but it may be good enough at MEDMOD'S level of aggregation and it would be relatively easy to change if a more rigorous approach were desired.

4. Damage to resources cannot be automatically repaired during a MEDMOD simulation. This limitation would be easy to remove, if desired, by adding code to allow the repair of resources during the simulation. However, it would be difficult to reflect repair rates which differ as a function of the real types of resources which are aggregated into the notional types played in MEDMOD. For example, it would not be hard to allow AEW escorts, on average, to be repaired at a different rate than average ASW escorts, but it would be hard to allow the various real types of AEW ships that make up the one notional type of AEW escort simulated in MEDMOD to be repaired at different rates. All resources except carriers can be "manually" repaired in MEDMOD using Subroutine TIMET (and NAVMOD will allow all resources including carriers to be repaired this way).

5. MEDMOD does not simulate Red aircraft carriers at the same level of detail as it simulates Blue aircraft carriers. A rough and implicit simulation of Red aircraft carriers is possible in MEDMOD, and NAVMOD will allow a much finer, but still implicit, simulation of Red aircraft carriers.

6. The number of Blue aircraft carriers, XPLAT, is constant in MEDMOD. This is not a significant limitation since the relative average effectiveness of the carriers, XEFFCM, is varied. However, in reality, a carrier might be sufficiently damaged that it leaves the task force thereby reducing the number of carriers (but increasing the average effectiveness of the remaining carriers).

7. Red bombers must be ordered by effectiveness in MEDMOD. That is, if multiple types of Red bombers are being simulated, then Red bombers of type 1 must be better than Red bombers of type 2, and so forth, where "better" is measured in terms of velocity and penetration ranges only. In particular, the input arrays VB, D1T, and D2T must satisfy:

$$VB(K) \geq VB(K+1)$$

$$D1T(I,K) \geq D1T(I,K+1) \qquad I=1,2$$

$$D2T(I,K) \geq D2T(I,K+1) \qquad I=1,2$$

for all K such that $1 \leq K \leq NKRB-1$.

8. Armed AEW aircraft are not simulated in MEDMOD. (The R-245 model simulates land-based, but not sea-based, armed AEW aircraft.)

3. Limitations Concerning Interactions

1. The various combat models (i.e., combat subroutines) of MEDMOD are at various levels of detail. In particular,

the most detailed combat model is the model of air and submarine attack on the task force in Subroutine CTFMOD. For example, CTFMOD explicitly distinguishes between killing a Red bomber and killing an ASM launched from a Red bomber, and, in doing so, it explicitly represents the self-defense capabilities of Blue ships against Red missiles. This level of detail is not contained in other subroutines of MEDMOD. Conversely, the D-day interaction simulated in Subroutine DDAY constitutes the least detailed combat model in MEDMOD. The idea here is that the first few hours of D-day combat can be more appropriately simulated "off-line" by a model with details and interactions specifically suited to D-day combat. The results of such a model could then be fed into the aggregated model of Subroutine D-Day. The other combat models of MEDMOD are roughly comparable in detail and lie between these two extremes. This structure was adopted because of the relative importance of the interactions in CTFMOD and because of the availability of other models to serve as combat models in MEDMOD. The point of this limitation is to note that some interactions simulated in CTFMOD have corresponding interactions that are simulated with less detail in other subroutines of MEDMOD, and it would be possible (though not necessarily desirable) to simulate these other interactions at the same level of detail as in CTFMOD.

2. MEDMOD will not automatically allow Red multi-purpose aircraft to serve as bombers on some days and as escort aircraft on other days.

3. The model of Blue air attacks on Red (land) airbases simulates attacks on aircraft in the open and in shelters, but it does not simulate attacks on runways or other facilities.

4. MEDMOD assumes coordination between air-launched and submarine-launched cruise missiles in that both types of cruise missiles contribute towards saturation of the area

and point defense of the task force (i.e., MEDMOD assumes that both types of missiles arrive at their targets at about the same time). If it is desired to explicitly simulate lack of coordination between air and submarine launches, then the current procedure would need to be modified. Appropriate modifications would not be difficult provided that the specification of which type of missile arrives first can be made. Damage due to the missiles that arrive first would have to be assessed before the later arriving missiles are engaged. (That is, later arriving missiles would not benefit from saturation due to earlier arriving missiles but would benefit from damage caused by earlier arriving missiles.) As a related point, saturation is played in a relatively simple way in MEDMOD. A more general and more sophisticated approach based on queueing theory has been developed and tested at IDA, but it has not been incorporated into MEDMOD. If improvements to this portion of MEDMOD are made, then this more general queueing theory approach to saturation could be considered.

5. All Blue surface ships and ASW aircraft that can shoot at Red submarines attacking the task force do so before those Red submarines can fire at the task force in Subroutine CTFMOD. (This restriction does not apply to Subroutine DDAY.)

6. After the Red submarines that are attacking the carrier task force have launched their weapons (missiles or torpedoes), they are assumed to escape the task force without further harm. That is, flaming datum prosecution of submarines is not simulated in MEDMOD.

7. Blue carrier-based aircraft can engage only Red aircraft, not ASMs launched from these aircraft.

8. Many tactical decisions are made in MEDMOD. These decisions typically involve when to attack, where to attack, how much to attack with, and similar questions for defense.

These decisions either are made directly by inputs or are made according to relatively simple decision rules based on input parameters. Decisions made directly by inputs do not (automatically) consider the status of the war at the time the decision is made. Decisions made by simple decision rules consider some aspects of the status of the war, but they are not so elaborate as to consider every related aspect of combat, nor do these decision rules solve for optimal decisions with respect to an overall measure of effectiveness. The intent in constructing these decision rules was to keep them relatively simple, to base decisions only on those aspects of combat that would usually be the most important, and to usually make reasonable but not necessarily optimal decisions. However, the fact that the decision rules used in MEDMOD are simple heuristic rules based on only a few of the relevant aspects of combat means that these rules will, in general, produce non-optimal decisions which, from time to time, may be far from optimal according to overall measures of effectiveness.

4. A Limitation Concerning Outputs

MEDMOD does not produce a summary killer-target (or killer-victim) scoreboard. Such a scoreboard could be manually constructed by examining the detailed output of MEDMOD, and additional code and computer variables could be added to MEDMOD to produce one or more such scoreboards, but none are automatically produced now.

REFERENCES

[1] Bracken, J., J.W. Graves, J.H. Grotte, R.H. Jakobovits, C.C. Weissman, *Cost-Effectiveness Evaluation of Alternative Carrier Task Forces* (U), IDA Report R-245, Institute for Defense Analyses, Arlington, VA, July 1979, SECRET.

[2] Buchanan, W.B., *et al., Sea Based Air Platform Cost/ Benefit Study* (U), CNS 1110, Center for Naval Analyses, Arlington, VA., January 1978, SECRET.

[3] Kerlin, Edward P., *et al., IDA TACNUC Model: Theater-Level Assessment of Conventional and Nuclear Combat,* Volume I: *Executive Summary;* and Volume II: *Detailed Description,* IDA Report R-211, Institute for Defense Analyses, Arlington, VA, October 1975.
Kerlin, Edward P., *et al., The IDA Tactical Warfare Model: A Theater-Level Model of Conventional, Nuclear, and Chemical Warfare,* Volume III: *Documentation;* Part I: *The Chemical Model and Other Modifications;* and Part II: *Program Description,* IDA Report R-211, Institute for Defense Analyses, Arlington, VA, November 1977.

[4] Anderson, L.B., P.A. Frazier, M.J. Hutzler, F.J. Smoot, *Documentation of the IDA Tactical Air Model (IDATAM) Computer Program,* IDA Paper P-1409, Institute for Defense Analyses, Arlington, VA, February 1979.

[5] Anderson, L.B., K. Gardner, J. Metzko, *IDA Combined Area Air Defense and Ship Self Defense Evaluation (IDACASE) Model,* Volume I: *Main Report;* Volume II: *Appendices A through D;* and Volume III: *Appendix E,* IDA Report R-255, Institute for Defense Analyses, Arlington, VA, August 1981.

[6] Karr, Alan F., *On a Class of Binomial Attrition Processes,* IDA Paper P-1031, Institute for Defense Analyses, Arlington, VA, June 1974.

[7] Anderson, L.B., *Some Concepts Concerning the Incorporation of Multiple Attrition Assessments and Night Combat into IDAGAM,* Working Paper WP-8 of IDA Project 3609, Institute for Defense Analyses, Arlington, VA, October 31, 1980.

[8] Schultis, William J., *et al., Net Assessment Methodologies and Critical Data Elements for Strategic and Theater Force Comparisons for Total Force Capability Assessment (TFCA) Interim Report* (U), IDA Paper P-1536, Institute for Defense Analyses, Arlington, VA, November 1980.

[9] Karr, Alan F., *Some Questions of Approximation Involving Lanchester and Binomial Attrition Processes,* Working Paper WP-10 of IDA Project 2371, Institute for Defense Analyses, Arlington, VA, January 1981.

[10] Anderson, L.B., J. Bracken, E.L. Schwartz, *Revised OPTSA Model* (in 3 vols.), Volume 1: *Methodology;* Volume 2: *Computer Program Documentation;* and Volume 3: *The OPTSA Print-Run Program,* IDA Paper P-1111, Institute for Defense Analyses, Arlington, VA, October 1975.

[11] Karr, Alan F., *Lanchester Attrition Processes and Theater-Level Combat Models,* IDA Paper P-1528, Institute for Defense Analyses, Arlington, VA, September 1981.

[12] Farrell, J.W., Memorandums to R.J. Hunt: "A Functional Description of the FLOATS 3A Model," FSE-047, dated December 2, 1974; "FLOATS 3A Input Specification," FSE-048, dated December 2, 1974; "FLOATS 3A Output Formats," FSE-049, dated December 2, 1974; "Description of FLOATS 3A Subroutines," FSE-050, dated December 30, 1974, Applied Physics Laboratory, The Johns Hopkins University, Silver Spring, MD.

[13] Evans, R.F., T. Hilands, R.V. Powers, *Design Notebook for Naval Air Defense Simulation (NADS),* Project Waterwheel, TRW Defense & Space Systems Group, McLean, VA, March 1981.

APPENDIX A

THE INP ROUTINE

CONTENTS OF APPENDIX A

FIGURES

TABLE

THE INP ROUTINE

I. INTRODUCTION

IDATAM is a model of air combat, and IDACASE is a model of naval combat. Both of these models use essentially the same computer routine to read and display inputs. This routine is called INP.[1]

This appendix contains extracts from the documentation of IDATAM (IDA Paper P-1409, February 1979) and the documentation of IDACASE (IDA Report R-255, September 1981). These extracts discuss: (1) the preparation of inputs in the format required by INP, (2) the steps required to add or delete input variables in INP, (3) the conversion of INP to computers other than those like IDA's CDC-6400, and (4) some of the features of INP.

The extracts from the documentation of IDATAM comprise Chapter II of this paper. These extracts consist of Chapter II (Sections B and C) and Appendix F of P-1409.

The extracts from the documentation of IDACASE (portions of Appendix C of that documentation) comprise Chapter III of this paper.

One difference between these two extracts is that IDACASE is a Monte Carlo model, and the first input to IDACASE is a card containing the seed (or a blank card, which causes a seed to be randomly selected). IDATAM is a deterministic model and so needs no seed. The format of the inputs to

---

[1]A somewhat different version of INP is used to read and display inputs for the TACWAR model (IDA Report R-211, Volume III, Part II, November 1977).

IDATAM is identical to the formats of the inputs to IDACASE, except that IDATAM has no seed card.

MEDMOD, like IDATAM, is a deterministic model, and so MEDMOD does not need (and cannot accept) the seed card described in the extract from IDACASE.

One feature of INP is that it allows "mid-run" changes to input variables to be read in INP; these changes are read, stored, and then applied during the computer run by an associated routine called TIMET. IDATAM models an air war as occurring over many time periods (e.g., over many days), and IDATAM uses TIMET to change input values during the course of the war (i.e., between specified time periods). IDACASE models one raid of Antiship Missiles (ASMs) attacking a naval task force, but it does not model multiple time periods. Accordingly, IDACASE cannot use the TIMET routine to update input variables over the course of a war. Instead, IDACASE uses TIMET to allow several different cases or scenarios to be evaluated in one computer run. MEDMOD, like IDATAM, models combat over multiple time periods and so it uses TIMET in a manner similar to IDATAM, not IDACASE. The format of the inputs used by TIMET is identical for IDATAM, IDACASE, and MEDMOD--IDACASE differs from IDATAM and MEDMOD in the interpretation and use of these inputs.

II. EXTRACTS FROM THE DOCUMENTATION OF IDATAM CONCERNING INP

This section consists of Sections B and C of Chapter II and of Appendix F of the documentation of IDATAM (IDA Paper P-1409).

B. MACHINE CONVERSION

IDATAM is written in FORTRAN for a CDC 6400 computer with 150K octal core. It can be converted to other machines which have a FORTRAN compiler. However, some changes may be required. This chapter outlines parts of the program that may need to be changed in order to run IDATAM on machines other than CDC computers.

Conventions of the CDC 6400 computer require the first card of a program to be a PROGRAM card. All files are declared in the PROGRAM card of MAIN. IDATAM is broken into overlays. By CDC convention, the first routine in the overlay must have the characteristics of a FORTRAN main program (not a subroutine). Therefore, there is a PROGRAM card for each overlay or a total of seven PROGRAM cards. Overlays are defined by an OVERLAY card with the following format:

$$\text{OVERLAY}(\text{lfn}, l_1, l_2)$$

where

lfn = the logical file name of the retention file, (i.e., IDATAM or ATTRTN)

l_1 = primary level number

l_2 = secondary level number.

Since there is an OVERLAY card for each overlay there are a total of seven OVERLAY cards. An overlay is called by the following statement:

$$\text{CALL OVERLAY }(\text{fn}, l_1, l_2, \text{p})$$

where

- fn - the logical file name of the retention file in left-justified hollerith code (i.e., 6HIDATAM, 6HATTRTN)
- l_1 - primary level number
- l_2 - secondary level number
- p - recall parameter. If p equals 6HRECALL, the overlay is not reloaded if it is in memory.

There are six OVERLAY calls in MAIN, and two in AIRATT making a total of eight CALL OVERLAY cards. This makes a total of 22 cards (7 PROGRAM, 7 OVERLAY, and 8 CALL OVERLAY) which may need to be changed.

The input routine, INP, was designed to assist the user. As a result, the input formats are general and easily understood, but the routine itself is fairly complicated. There are three main concerns in INP in relation to machine conversion: ENCODE/DECODE statements, word size, and character conversion.

ENCODE/DECODE are statements which perform memory-to-memory transfer of data often called core-to-core I/O. The parameters are defined as follows:

ENCODE(c,n,v)L

where

- c - unsigned integer constant or a simple integer variable (not subscripted) specifying the number of characters in the record; c may be an arbitrary number of BCD characters
- n - statement number, variable identifier, or formal parameters representing the FORTRAN statement
- v - variable or array identifier which supplies the starting location of the BCD record
- L - input/output list.

The information in the list variables, L, is transmitted according to the FORMAT statement n and stored in locations starting at v, c, BCD characters per record.

DECODE(c,n,v)L .

The information in c consecutive BCD characters (starting at address v) is transmitted according to the FORMAT statement n and stored in the list variables.

The CDC 6400 computer has a 60 bit word, 6 bits per character making a 10 character word. Variables which need this character space will have to be declared large enough to handle up to 10 characters and be referenced accordingly.

As will be explained in Section C below, there is an option in the input routine that allows the user to increment or replace input data after any cycle. To implement this option there is a two-character cycle code in columns 19-20 of the input data cards. Cycles 0 through 99 are coded as required and blanks are converted to 0. If a simulation is longer than 99 days, day 100 is coded as A0, 101 as A1, 110 as B0, 197 as J7, etc. This two character code is read in A format and converted to a numeric value in machine dependent code. Alphanumeric characters are assumed to have the following values:

Character	Octal Code	Character	Octal Code
A	01	0	33
B	02	1	34
C	03	2	35
.	.	.	.
.	.	.	.
.	.	.	.
Z	32	9	44

It should be easy to make this conversion for any machine.

C. PREPARATION OF INPUTS

There are two consecutive groups of data which are read in as inputs. The first is an alphabetically ordered deck of definitions of the input variables. The second group of data consists of the values of the input variables. Each deck is followed by a card with "ZZZZZZ" in the first six columns as a delimiter.

The definition of the input deck should not change unless an input's definition is altered. The format of the cards in this deck is:

Card Column	Contents
1-6	Input variables name (left justified)
7	Sequence number of card (1-5)
8-77	Definition of the input variable.

Up to five cards may be used per definition and the sequence number must have the value one through five.

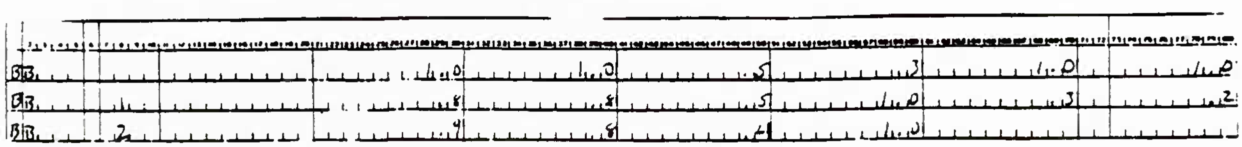
The card format for the second deck is:

Card Column	Contents
1-6	Input variables name (left justified)
7-8	Continuation code
9	Not used
10-12	First argument, if needed, (right justified)
13-15	Second argument, if needed, (right justified)
16-18	Third argument, if needed, (right justified)
19-20	Cycle number, if needed, (right justified).

Card Column	Contents
21-30	Data field 1
31-40	Data field 2
41-50	Data field 3
51-60	Data field 4
61-70	Data field 5
71-80	Data field 6

There are six ten-character data fields (columns 21-80) for actual data values. All inputs follow the standard FORTRAN typing rule (i.e., names beginning with I-N are integer, and otherwise are assumed to be real). Floating point numbers are read in with an F10.0 format. This means a decimal point must appear in the field if the input value has a fractional part. Integers are read in with an I10 format and must be right justified. There are also four alphanumeric variables, AACT, AAMT, ALRS, and AMRS, whose values are read in with an A10 format.

The first, second, and third arguments are used to indicate how the data are to be stored. For example, assume input values are to be coded for a one-dimensional array, BB, dimensioned to 16. Three cards are required. The first card contains BB in columns 1 and 2, either a blank or 0 in column 8, and the data values for BB(1) through BB(6) in the six data fields. The second card contains the variable name, a "1" in column 8, and has values for locations BB(7) through BB(12). The third card contains the variable name, a "2" in column 8, and has values for locations 13-16 of BB.



BB				1.0	1.0	.5	3	1.0	1.0
BB		1		8	.8	.5	1.0	3	2
BB		2		.9	8	4	1.0		

Assume input is required for an array ABCDEF(2,7) with the following values:

		Column						
		1	2	3	4	5	6	7
Row	1	29	140	11.4	1.0	0.5	3.5	10
	2	10	10.5	11.4	10.5	12	12	13

The coding can be done in two ways. The first way is to enter the data in rows. Specify the variable name, the first argument, I=1, and enter 6 values on the first card, followed by a second card with the variable name, a "1" in column 8, I=1, and a single value. Then repeat the above for I=2.

ABCDEF		1	29	140	11.4	1.0	.5	3.5
ABCDEF	1	1	10					
ABCDEF		2	10	10.5	11.4	10.5	12	12
ABCDEF	1	2	13					

The second method is to specify the second argument, J=1, on a card and provide two values; repeat for J=1 through 7.

ABCDEF	1	29	1.0
ABCDEF	2	140	10.5
ABCDEF	3	11.4	11.4
ABCDEF	4	1.0	10.5
ABCDEF	5	.5	12
ABCDEF	6	3.5	12
ABCDEF	7	10	13

The latter method enters the data by columns. For this example, the first method requires fewer cards, but the user may find the second method easier to read, verify, and if necessary, change.

Next, assume input is required for array MMM(2,6,2) with the values:

				Column					
				1	2	3	4	5	6
Plane	1	Row	1	1	3	5	7	9	11
			2	2	4	6	8	10	12
	2	Row	1	13	15	17	19	21	23
			2	14	16	18	20	22	24

In this case three methods are available. Either specify arguments I and J, J and K, or I and K. The first two methods will require twelve cards, whereas the third method will require four cards. If I and J are specified, the twelve cards would appear as:

MAA		1 1	1	13
MAA		1 2	3	15
MAA		1 3	5	17
MAA		1 4	7	19
MAA		1 5	9	21
MAA		1 6	11	23
MAA		2 1	2	14
MAA		2 2	4	16
MAA		2 3	6	18
MAA		2 4	8	20
MAA		2 5	10	22
MAA		2 6	12	24

If I and K are specified, the four cards would appear as:

MAA		1 1	1	3	5	7	9		11
MAA		1 2	13	15	17	19	21		23
MAA		2 1	2	4	6	8	10		12
MAA		2 2	14	16	18	20	22		24

Once an input deck has been set up, the user may wish to make variations by changing the values for some of the input variables. One way to do this without having to delete the original card(s) is to place the card(s) with the updated

value(s) anywhere after the original card(s), but before the input cards for the next cycle. For cycle zero, the program simply accepts the last input value. The same principle holds for replacement variables for cycles beyond cycle zero. However, all cards for incremental variables are processed after cycle zero.

For example, if it is desired to change the fourth element of array BB to 10.0, the original card could be retyped with 10.0 in the fourth data field, or an additional card could be inserted anywhere after the original cards. This card would contain "BB" in columns 1 and 2, a "4" in column 12, and the value "10.0" in data field 1, columns 21-30. This latter method may be convenient if there are many changes to be made.

Input variables can be updated after any cycle. The cycle number is in card columns 19-20. As explained at the end of Section B above, cycles zero through ninety-nine are represented by the numbers 0 through 99, with 0 or a blank representing cycle 0. If the simulation is to be run longer than 99 cycles, A0 represents 100 cycles, A1 represents 101, B0 represents 110 cycles, J7 represents 197 cycles, etc. Cards in this deck must be in ascending order of cycles. Some input variables are incremented (see Appendix C [of P-1409], Incremental Variables) by the values in the data fields; all values are replaced by the new values.

The program [IDATAM] is now dimensioned so that the upper bounds on the following variables are:

NAC	(maximum number of types of aircraft)	15
NAM	(maximum number of types of air munitions)	10
NLS	(maximum number of types of long range SAMs/HIMADS)	3
NMS	(maximum number of types of medium range SAMs/SHORADS)	6
NQRAT	(maximum number of types of QRA aircraft)	7
NS	(maximum number of sectors)	2

These variables can be increased provided that all the appropriate COMMON and DIMENSION statements are changed, and that the program will still fit into core.

There are no internal checks made on the consistency of inputs. For instance, there will be problems if the variables listed in Appendix D [of P-1409] are zero. Care should also be taken when preparing the variables which involve the allocations of aircraft.

PROCEDURE FOR ALTERING THE LIST OF INPUT VARIABLES

This appendix is only for the user who wishes to add, delete, or redimension input variables in the model. The data decks for IDATAM are relatively easy to prepare. However, the input routine, INP, which processes these decks is fairly complex. To allow the user the convenience of a simplified input scheme, a cross reference map of the input variables, which are defined in blank common, is used in INP. The cross reference map, IVARQ, is keyed on the input variable name. If blank common is to be changed, IVARQ must be updated to reflect the change. IVARQ is defined by a set of DATA statements. An independent program named COMM is used to recreate the DATA statements when COMMON is changed. A run to change a statement in COMMON from "COMMON BLRS(3,2,2)" to "COMMON BLRS(4,2,2)" might appear as:

Card No.	Card	Description of Action
1	JOB CARD	Job identifier, request 150K.
2	REQUEST(OLDPL,HI)ɓ(0001/FP)	Request tape which contains program.
3	NUPDATE(N=PL)	Update COMMON (cards 21-23).
4	RETURN OLDPL	Returns OLDPL.
5	NUPDATE(Q,P=PL)	Copy updated COMMON onto TAPE 10 (cards 25-27).[1]
6	FTN.	Compile Program COMM (noted card 29).

[1]See the text following this example for a specification of TAPE 10.

Card No.	Card	Description of Action
7	LGO.	Load and execute Program COMM (data noted card 21)(decreases field length).
8	REWIND,TAPE15.	Rewind tape 15 for later processing.
9	RFL,150000.	Increase field length to 150K.
10	REQUEST(NEWPL,HI)⌀(SAVE)	Request a save tape named NEWPL.
11	NUPDATE(N,F,R=C,P=PL)	Do update (cards 33-35) to create a new program on NEWPL.
12	REWIND LGO.	Rewind the load file.
13	FTN(I=COMPILE,A,T,R=3)	Compile program to create load file, LGO.
14	REQUEST(BIN,HI)⌀(SAVE)	Request a save tape named BIN.
15	REWIND,LGO	Rewind the load file.
16	COPBF(LGO,BIN)	Copy LGO onto BIN.
17	REWIND,LGO.	Rewind load file.
18	CLEAR	Zero out memory.
19	LGO.	Load and execute program, IDATAM.
		NOTE:End of Control Stream
20	7/8/9	Delimiter.
21	*IDENT COMCHG.1	Arbitrary identifier for NUPDATE.[1]
22	*DELETE COMM.13	Delete the old COMMON card, COMMON BLRS(3,2,2).[2]
23	COMMON BLRS(4,2,2)[3]	Insert the new card in COMMON.

[1]COMCHG.1 is the example identifier used here.

[2]COMM.13 is assumed to specify the card COMMON BLRS(3,2,2) in this example.

[3]This card starts in column 7.

Card No.	Card	Description of Action
24	7/8/9	Delimiter.
25	*IDENT COPY1	Arbitrary identifier for NUPDATE.[4]
26	*COMPILE COMM[5]	Compile COMMON.
27	*COPY COMM,COMM.2, CHG13.21,TAPE10	Copy updated COMMON onto TAPE10.[6]
28	7/8/9	Delimiter.
29s	PROGRAM COMM[7]	Create new data statements on TAPE15 from TAPE10.
30	7/8/9	Delimiter.
31s	DATA FOR PROGRAM COMM[8]	Exceptions to type rule.[8]
32	7/8/9	Delimiter.
33	*IDENT DSTMT2[9]	Arbitrary identifier for NUPDATE.
34	*YANK DSTMT1[9]	Delete old data statements.
35	*INSERT CHG18.6[10]	Insert new data statements.
36	*READ TAPE15	Read the data statements from TAPE15.
37	7/8/9	Delimiter.
38	IDATAM DATA DECKS	Definitions and values.
39	6/7/8/9	Delimiter.

---

[5]COMM refers to a deck, not a Program here.

[6]COMM.2 and CHG13.21 are assumed to include all of the inputs and nothing else.

[7]The FORTRAN deck of cards which is PROGRAM COMM goes here.

[8]The required cards which give the data for PROGRAM COMM go here—see the text which follows for the definition of "exceptions to type rule."

[9]This assumes that the old data statements were all identified by DSTMT1 and that the new set of data statements is to be identified by DSTMT2.

[10]This assumes that the identifier of the card just before the old data statements (i.e., just before DSTMT1.1) is CHG18.6.

This input scheme uses NUPDATE, a program maintenance routine, to update IDATAM. If your system does not support a program maintenance routine, change program COMM to punch the new data statements and insert them by hand.

The data input to program COMM is the COMMON deck (assumed to reside on TAPE10) followed by exceptions noted on cards in the data deck.[1] The exceptions to be noted are with respect to the "typing of variables" and the updating of input variables in TIMET. The conventions for typing of variables is the standard for FORTRAN, i.e., names beginning with I-N are integer, and otherwise assumed real. If a variable contains alphanumeric information, it must be noted as "ALPHA". If a variable is to be accepted as real when it begins with I-N, it must be noted as "REAL". Similarly, a variable that does not begin with I-N (but is typed integer) must be noted as "INTEGER". Processing a variable at time t is always assumed to be "replacement". If the variable is to be incremented, it must be noted as "INCREM". If a variable is to be treated as "side-implicit" it should be noted as "BLURED". (Note: the TACWAR model, currently being developed at IDA, uses this option, IDATAM does not.) Following an end of file on TAPE10, input for variable types is expected from the system's card reader. This input is free format in columns 6-72. Column 6 is any non-blank character denoting continuation. The first string of characters to be input is ",END," to terminate the reading of COMMON. Exceptions are then input as strings such as "OPER,V_1,V_2,...,V_n". OPER can have the values ALPHA, INTEGER, REAL, INCREM, or BLURED. The values of V_1 are variable names.

---

[1]Since MEDMOD uses TAPE10 for a different purpose, a different file should be used here if this fully automated procedure is to be employed with MEDMOD. Alternatively, the three-step procedure described in Section III.D.2 of this appendix can be (and has been) used with MEDMOD.

The exceptions are terminated by the string "END,". The following may be a helpful example:

```
* ,END,ALPHA,AACT,AAMT,ALRS,AMRS,
* INCREM,ACFS,ACRS,INTEGER,XYZ,
* END,
```

No exceptions need be included.

III. EXTRACTS FROM THE DOCUMENTATION OF IDACASE CONCERNING INP

This section consists of portions of Appendix C of the documentation of IDACASE (IDA Report R-255).

PREPARATION OF INPUTS

A. SAMPLE INPUT FILE

Figure C-1 shows a sample deck of input data used to run IDACASE. This deck was used to generate Figure 28. The type of data needed is discussed next.

B. INPUT RECORDS

1. The Input Data Deck

The three types of input data records provided to IDACASE in the input card deck are:

1. Seed card
2. Input variable definitions
3. Input variable values.

Figure C-1 shows a sample working data deck which illustrates the three types of inputs. This input deck consists of (1) a seed card, followed by (2) a deck of input definition cards ended with an input definition card with a "ZZZZZZ" for variable name, followed by (3) a deck of input value cards ended with a "ZZZZZZ" for variable name.

```
0000000012=23121663
AENOBP1(K,IBP) MAXIMUM ELEVATION ANGLE NUMBER IBP FOR AREA OF DETECTION    D
AENOBP2BY SHIP K AGAINST HIGH DIVER OR DIVING SEA SKIMMER--DEG.
AHDDA 1DIVE ANGLE ASSUMED BY FLEET FOR ALL HIGH DIVERS, MEASURED FROM      ASM
AHDDA 2ASSUMED FLIGHT PATH DURING DIVE TO ITS HORIZONTAL PROJECTION--DEG.
ANGJMN1(J)AAD SS/FCC J MINIMUM FIRING BEARING, ANGLE, WITH VERTEX AT       AD
ANGJMN2FIRING SHIP, MEASURED IN HORIZONTAL PLANE FROM FLEET 0 DEG BEARING
ANGJMN3TO POINT UNDER ASM--DEG.
ANGJMX1(J)AAD SS/FCC J MAXIMUM FIRING BEARING, ANGLE, WITH VERTEX AT       AD
ANGJMX2FIRING SHIP, MEASURED IN HORIZONTAL PLANE FROM FLEET 0 DEG BEARING
ANGJMX3TO POINT UNDER ASM--DEG.
ASMROL1(I)HORIZONTAL RANGE FROM TARGET SHIP TO ASM I LAUNCH--FT.           ASM
CBR1C11(JU)MINIMUM FIRING BEARING FOR SSD UNGUIDED SYSTEM JU, MEASURED     SD
CBR1C12COUNTER-CLOCKWISE IN HORIZONTAL PLANE FROM SHIPS BOW--DEG.
CBR2C11(JU)MAXIMUM FIRING BEARING FOR SSD UNGUIDED SYSTEM JU, MEASURED     SD
CBR2C12COUNTER-CLOCKWISE IN HORIZONTAL PLANE FROM SHIPS BOW--DEG.
CDWNDS1STEPSIZE FOR DOWNRANGE DISTANCE, ESCORT TO TARGET SHIP, IN AAD      CNT
CDWNDS2ENGAGEMENT CONTOUR TABLES FOR DIVING SEA SKIMMERS--FT.
CDWNR 1STEPSIZE FOR DOWNRANGE DISTANCE, ESCORT TO TARGET SHIP, IN AAD      CNT
CDWNR 2ENGAGEMENT CONTOUR TABLES FOR HIGH DIVERS--FT.
CFANG 1(K) SHIP I TO SHIP K ANGLE COMPARED TO 0 DEG FOR FLEET--DEG.        S
CFRANG1(K) SHIP I TO SHIP K RANGE--FT.
CMAXC11(JUTYP)MAXIMUM HORIZONTAL RANGE FOR INTERCEPT BY UNGUIDED SSD TYPE  SD
CMAXC12JUTYP--FT.
CMINC11(JUTYP)MINIMUM HORIZONTAL RANGE FOR INTERCEPT BY UNGUIDED SSD TYPE  SD
CMINC12JUTYP--FT.
COFFDS1STEPSIZE FOR OFFSET RANGE OF ESCORT FROM TARGET SHIP IN AAD         CNT
COFFDS2ENGAGEMENT CONTOUR TABLES FOR DIVING SEA SKIMMERS--FT.
COFFR 1STEPSIZE FOR OFFSET RANGE OF ESCORT FROM TARGET SHIP IN AAD         CNT
COFFR 2ENGAGEMENT CONTOUR TABLES FOR HIGH DIVERS--FT.
COUR  1(I)ACTUAL COURSE RELATIVE TO TARGET SHIP OF ASM I -- ANGLE, WITH ASM
COUR  2VERTEX AT TARGET SHIP, MEASURED IN HORIZONTAL PLANE FROM FLEET
COUR  30 DEG TO POINT UNDER ASM--DEG.
CWAVE 1(IMT)CONSTANT BY WHICH NEW WAVE IS TO BE SHIFTED FOR ASM TYPE IMT.  ASM
CWAVE 2TIME FOR WAVE N IS TIME FOR WAVE N-1 PLUS FLYIN TIME (SEE INTWAV)
CWAVE 3PLUS CWAVE(IMT)--SEC.
DETMIN1(K)MINIMUM DELTA TIME ASM MUST REMAIN IN SHIP K DETECTION (TINDET   D
DETMIN2TO TOUTDT) FOR VALID DETECTION--SEC.
DISBP 1(K,IBP)DETECTION RANGE NUMBER IBP FOR DETECTING SHIP K AGAINST      D
DISBP 2HIGH DIVERS OR DIVING SEA SKIMMERS--FT.
DISSCL1(K,IS)FACTOR FOR SCALING DETECTION RANGES (DISBP) OF SHIP K FOR     D
DISSCL2JAMMING IN SECTOR IS.
DRLDSL1HORIZONTAL DISTANCE FOR DIVING SEA SKIMMER LAUNCH RANGE TO RANGE    ASM
DRLDSL2OF LEVELING OFF AND STARTING SKIMMING--FT.
DRNGMN1(IMT)MINIMUM HORIZONTAL RANGE FOR DETECTION OF ASM TYPE IMT--FT.    D
DRNGMX1(IMT)MAXIMUM HORIZONTAL RANGE FOR DETECTION OF ASM TYPE IMT--FT.    D
DSCA  1CRUISE ALTITUDE BEFORE DIVE FOR DIVING SEA SKIMMER--FT.             ASM
DSDVEL1DIVING SEA SKIMMER HORIZONTAL VELOCITY DURING AND BEFORE DIVE--FPS  ASM
DSLKTS1DISTANCE FROM RANGE OF LEVELING OFF TO RANGE TARGET SHIP IS KNOWN   ASM
DSLKTS2TO FLEET FOR DIVING SEA SKIMMER--FT.
DSSVEL1DIVING SEA SKIMMER VELOCITY AFTER DIVE (DURING SKIMMING)--FPS.      ASM
DTAFTH1(J)DELTA TIME FROM FIRING (FOR ACTIVE SYSTEMS) OR FROM INTERCEPT    AD
DTAFTH2(FOR SAM/SAT SYSTEMS) TILL SS/FCC AVAILABLE FOR NEW AAD ASSIGNMENT
DTAFTH3--SEC.
DTBRSU1(JGTYP)TIME BETWEEN ROUNDS OF A SALVO FOR GUIDED SSD TYPE           SD
DTBRSU2JGTYP--SEC.
DTBSAL1(J)TIME BETWEEN ROUNDS OF A SALVO FOR AAD SS/FCC J--SEC.            AD
DTDET 1DELTA TIME FOR DETECTION TIME STEPS--SEC.                           D
DTDLC11(JUTYP)DELTA TIME FROM DETECTION TO FIRE FOR UNGUIDED SSD SYSTEM    SD
DTDLC12TYPE JUTYP--SEC.
DTDLSU1(JGTYP)DELTA TIME FROM DETECTION TO LAUNCH FOR GUIDED SSD SYSTEM    SD
DTDLSU2TYPE JGTYP--SEC.
DTHOC11(JGTYP)DELTA TIME FOR ADDITIONAL DELAY BEFORE CIWS FIRING DUE TO    SD
```

(Continued)

Figure C-1. LIST OF INPUT DATA CARDS

•

•

•

```
SEPKSU1(JGTYP,IMT)SINGLE SHOT PK FOR GUIDED SSD SYSTEM TYPE JGTYP          SD
SEPKSU2AGAINST ASM TYPE IMT.
TOCTIE1(J)RANKING OF AAD SS/FCRS FOR DECIDING ORDER OF CSF EVENTS          AD
TOCTIE2OCCURING AT IDENTIAL TIMES.
TOLINP1(I)CLOCK TIMES USED AS PLANNED TIME OF LAUNCH OF ASM I. THIS TIME   ASM
TOLINP2IS USED IN GENERATION OF NORMALLY DISTRIBUTED TIMES OF
TOLINP3LAUNCH--SEC.
TOLPAM1(IMT,IST)STATISTICAL PARAMETERS OF NORMAL DISTRIBUTION FOR TIME OF ASM
TOLPAM2LAUNCH BASED ON PLANNED LAUNCH TIME FOR ASM TYPE IMI (IST = 1 FOR
TOLPAM3STANDARD DEVIATION, IST = 2 FOR DELTA MEAN)--SEC.
VGUNSU1(JGTYP)AVERAGE HORIZONTAL VELOCITY OF ROUND FROM GUIDED SSD         SD
VGUNSU2SYSTEM TYPE JGTYP--FPS.
VONOCI1(JUTYP)AVERAGE HORIZONTAL VELOCITY OF ROUND FROM UNGUIDED SSD       SD
VONOCI2SYSTEM TYPE JUTYP--FPS
ZZZZZZ
AFNOBR     1          20.        40.
AFNOBR     2          20.        60.
AFNOBR     3          20.        40.
AFNOBR     4          20.        60.
AFNOBR     5          20.        40.
AFNOBR     6          20.        60.
AFNOBR     7          20.        40.
AHDDA                 40.
ANGJMN 0              260.       240.       76.        76.        20.        20.
ANGJMN 1              20.        20.        20.        20.        270.       270.
ANGJMN 2              90.        90.        30.        30.        20.        20.
ANGJMN 3              20.
ANGJMX 0              100.       100.       284.       284.       340.       340.
ANGJMX 1              340.       340.       340.       340.       90.        90.
ANGJMX 2              270.       270.       330.       330.       340.       340.
ANGJMX 3              340.
AEMROL 0              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 1              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 2              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 3              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 4              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 5              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 6              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 7              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 8              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL 9              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL10              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL11              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL12              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL13              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL14              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL15              900000.    900000.    900000.    900000.    900000.    900000.
AEMROL16              900000.    900000.    900000.    900000.
```

•

•

(Continued)

•

Figure C-1. (Continued)

```
TALPAH          2      0.        ?.
VRUNSU                     1200       1850       1200       1800
VPNOCI                     3500
NUMWAV                        3          4          1          2          5
NASMS4          1            20         20         20
NASMS4          2            20         15         10
NASMS4          3            60          0          0
NASMS4          4            20          0         20
NASMS4          5             8          8          8
NCASE                         3
AFNORH  0       1     125.       50.        25.        25.        25.        25.
AFNORH  1       1     125.
AHODA                 230.
AFNORH  0       1     220.       20.        20.        20.        20.        20.
AFNORH  1       1     220.
ZZZZZZ
```

For brevity, portions of this figure have been deleted in this extract.

Figure C-1. (Concluded)

2. <u>The Seed Card</u>

The first card of the input deck is used to initialize the seed for random number generation. A sample seed record is shown in Figure C-2. The format for this card is:

Card Column	Contents
1-20	An octal number seen (use digits 0 through 7 only).

Specifying the same seed input enables the user to reproduce a random number stream in different computer runs. A blank or zero input for the seed causes random initialization (using the system clock) of the random number generator.

3. <u>Input Variable Definition Cards</u>

The deck of input variable definitions gives descriptive information concerning each variable which, for IDACASE, includes variable name, index name(s), and a variable definition. These definitions appear in the complete output of input variables and values provided by IDACASE's input routine.

A sample input variable definition record is given in Figure C-3. These variable definition cards must be in alphabetical order by variable name to be used, although omission or misordering of definitions will not affect data values.

Figure C-2. INPUT SEED CARD FORMAT

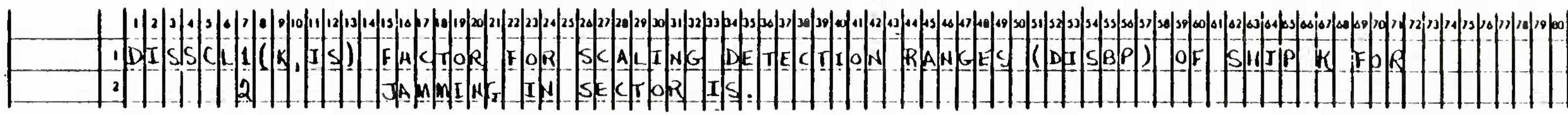


Figure C-3. INPUT VARIABLE DEFINITION CARD FORMAT

The format for the input variable definition card is:

Card Column	Contents
1-6	Input variable name (left justified).
7	Sequence number of card (1-5).
8-77	Definition of input variable.

Up to five cards may be used in a definition, and the sequence number must have a value from 1 to 5.

4. Input Variable Value Cards

The deck of input data values contains values to be assigned to selected input variables. These cards allow specification of variable name, index value(s), variable value(s), and case number for an input variable. All input variables are set to zero (by the input routine) unless values are entered with input variable value cards. The cards may be in any order within a Case (or cycle). (Note that both the variable definition and variable value decks can be sorted into useful or rational order.) If a card for a particular value for a variable is repeated in a Case, the last entry is used by the program.

The format for the input variable value card is:

Card Column	Contents
1-6	Input variable name (left justified)
7-8	Continuation code (integer)
9	Blank

10-12	First index value, if needed (right justified integer)
13-15	Second index value, if needed (right justified integer)
16-18	Third index value, if needed (right justified integer)
19-20	Cycle number = Case number less 1 (integer)
21-30	Data value 1
31-40	Data value 2
41-50	Data value 3
51-60	Data value 4
61-70	Data value 5
71-80	Data value 6

Sample input value records are given in Figure C-4. Inputs follow the FORTRAN typing rules and are read with FORTRAN format conventions. That is, real numbers are to be specified with F10.0 format while integers must be given in I10 format. There are also two alphanumeric variables, NSYSSD and NSYSCI, whose values must be provided in A10 format.

The index values in columns 10 through 18 allow flexibility in the entry of data values into arrays. With these inputs, indices of a dimensioned variable may be set. Only one index may be allowed to vary (by not specifying index values). The sample in Figure C-4 show examples of data entry for an undimensioned integer variable, NDS, and for real dimensioned variables DISBP(7,5) and PKCUM(10,2,5). The two examples for the two-dimensional variable DISBP(I,J) will give the same results for stored variable values.

Since only six values can be specified on an input card, the continuation code provides for additional (more than six)

```
NDS                         10


DISBP      1            290000    110000
DISBP      2            290000    110000
DISBP      3            290000    110000
DISBP      4            290000    110000
DISBP      5            290000    110000
DISBP      6            290000    110000
DISBP      7            290000    110000

DISBP  0      1         290000    290000    290000    290000    290000    290000
DISBP  1      1         290000
DISBP  0      2         110000    110000    110000    110000    110000    110000
DISBP  1      2         110000


PKCUM         1  1  .95       .94       .92       .91       .89       .865
PKCUM  1      1  1  .835      .785      .745      0.
PKCUM         2  1  1.        1.        .99       .98       .97       .965
PKCUM  1      2  1  .95       0.
```

Figure C-4. INPUT VARIABLE VALUE CARD FORMAT

values of a variable, for particular index values, to be given. The two samples of data input in Figure C-3 for DISBP and PKCUM show the use of the continuation code and index specification. The continuation code value is zero (or blank) for the first set of six values and increases by one for each succeeding set of six values. (All continuations need not be specified. If only the last 10 values of variable COUR (100) are to be specified, only the COUR cards with continuation index values 15 and 16 need be entered.)

The cycle number field of the input variable value card is used in IDACASE to allow several Cases to be included in a single computer run. The cycle number for the first Case is zero (or may be omitted) and subsequent cycle numbers increase by one. (That is, Case 2 corresponds to cycle 1, Case 3 to cycle 2, and so on.) Input variable values must be grouped by cycle number and ordered by increasing cycle number, and the input variale NCASE must be set to the number of Cases to be run.

C. OUTPUT DISPLAY OF INPUT VARIABLES

1. Complete Input Display by INP

The initial output display provided by IDACASE is a listing of all input variable names, definitions and values provided by the input subroutine, INP. The first page of this output is shown in Figure C-5. This page of output shows the seed input in the first line, followed by "TIME-T" prints which give input variables with values to be used in additional Cases. Below the "TIME-T" displays, the complete printout of definitions and values for all inputs for the first Case begins.

"TIME-T" prints are only given for variables to be changed after the first case. (The name "TIME-T", as in the subroutine TIMET, comes from application of this capability in earlier models to bringing in new data at certain times in a simulation.)

```
INPUT SEED = 000000000012523121663
SEED FOR RUN=000000000012523121663


TIME-T=    1       VARIABLE AENDBP 0     0  1   0---VALUES ARE BELOW
        25.00       25.00       25.00       25.00       25.00       25.00
TIME-T=    1       VARIABLE AENDBP 1     0  1   0---VALUES ARE BELOW
        25.00       0.          0.          0.          0.          0.
TIME-T=    2       VARIABLE AHDDA  0     0  0   0---VALUES ARE BELOW
        30.00       0.          0.          0.          0.          0.
TIME-T=    2       VARIABLE AENDBP 0     0  1   0---VALUES ARE BELOW
        20.00       20.00       20.00       20.00       20.00       20.00
TIME-T=    2       VARIABLE AENDBP 1     0  1   0---VALUES ARE BELOW
        20.00       0.          0.          0.          0.          0.
VARIABLE ---- AENDBP(  7,  5,  0)
                                (K,IBP) MAXIMUM ELEVATION ANGLE NUMBER IBP FOR AREA OF DETECTION   0
                                BY SHIP K AGAINST HIGH DIVER OR DIVING SEA SKIMMER--DEG.
        20.00       60.00       0.          0.          0.
        20.00       60.00       0.          0.          0.
        20.00       60.00       0.          0.          0.
        20.00       60.00       0.          0.          0.
        20.00       60.00       0.          0.          0.
        20.00       60.00       0.          0.          0.
        20.00       60.00       0.          0.          0.
VARIABLE ---- AHDDA (  1,  0,  0)
                                DIVE ANGLE ASSUMED BY FLEET FOR ALL HIGH DIVERS, MEASURED FROM      ASM
                                ASSUMED FLIGHT PATH DURING DIVE TO ITS HORIZONTAL PROJECTION--DEG.
        40.00
VARIABLE ---- ANGJMN( 40,  0,  0)
                                (J)AAD SS/FCC J MINIMUM FIRING BEARING, ANGLE, WITH VERTEX AT       AD
                                FIRING SHIP, MEASURED IN HORIZONTAL PLANE FROM FLEET 0 DEG BEARING
                                TO POINT UNDER ASM--DEG.
        260.0       260.0       76.00       76.00       20.00       20.00       20.00       20.00       20.00       20.00
        270.0       270.0       90.00       90.00       30.00       30.00       20.00       20.00       20.00       0.
        0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
        0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
VARIABLE ---- ANGJMX( 40,  0,  0)
                                (J)AAD SS/FCC J MAXIMUM FIRING BEARING, ANGLE, WITH VERTEX AT       AD
                                FIRING SHIP, MEASURED IN HORIZONTAL PLANE FROM FLEET 0 DEG BEARING
                                TO POINT UNDER ASM--DEG.
        100.0       100.0       284.0       284.0       340.0       340.0       340.0       340.0       340.0       340.0
        90.00       90.00       270.0       270.0       330.0       330.0       340.0       340.0       340.0       0.
        0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
        0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
VARIABLE ---- ASHROL(100,  0,  0)
                                (I)HORIZONTAL RANGE FROM TARGET SHIP TO ASM I LAUNCH--FT.          ASM
        .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06
        .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06
        .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06
        .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06
        .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06   .9000E+06
```

Figure C-5. SAMPLE OF COMPLETE DISPLAY OF INPUT DATA

"TIME-T 1" prints give seven new AENDBP(K,L) values for each ship (K) for Case 2, values of 25 compared to 20 given below for Case 1. "TIME-T 2" returns AENDBP to its original (Case 1) values and changes the value for AHDDA from 40. to 20.

2. Case Input Display by Function

A display of tables of inputs by function is the first output given for each IDACASE Case.

> This output is not provided by INP (it is provided by an output routine constructed specifically for IDACASE), and so the discussion of it and the corresponding figure (Figure C-6) are not included in this extract from the documentation of IDACASE.

D. THE INPUT ROUTINE INP

The IDACASE input routine, INP, allows flexible data input, as described above, by means of a complex FORTRAN subroutine. While this routine utilizes special capabilities of the CDC 6400, it can be (and has been) converted to run on other computers. Routine INP has been utilized and documented at IDA in conjunction with its inclusion in several models (in particular, see IDA Paper P-1409[2]).

[2] Anderson, L.B., P.A. Frazier, M.J. Hutzler, and F.J. Smoot. Documentation of the IDA Tactical Air Model (IDATAM) Computer Program. IDA Paper P-1409, Institute for Defense Analyses, Arlington, VA., February 1979.

1. <u>The INP Subroutine and Table IVARQ</u>

A table with information on input variables is included in INP to allow for flexible input format. This table provides the routine with basic input variable information needed for input processing in FORTRAN language. This information includes variable name, type, dimensions, and common block location. The table is stored, with DATA statements, in array (IVARQ(JV,IND) where JV indicates a variable (JV = 1 for the first variable in alphabetical order, 2 for the second, etc.) and IND is the index for types of information about variable JV, as shown in Table C-1. Figure C-7 shows sample COMMON statements with corresponding DATA statements describing the input variables in common necessary for INP operation.

2. <u>Changes in Input Variables</u>

The need to provide INP with a table of information about input variables requires production of a new IVARQ table whenever changes are made in the (1) number, (2) names, (3) dimensions, or (4) order in common of input variables. The information in IVARQ (DATA statements) must correspond to specifications for input variables which are stored in blank common, as shown in Figure C-7. A change in an input variable (in blank common) will necessitate:

(1) Producing a new set of IVARQ DATA statements,[1]

(2) Replacing old DATA statements for IVARQ in in INP with new DATA statements,

(3) Changing line(s) in common statements in the many routines containing the blank common block.

A complete new set of DATA statements is usually necessary (and simpler) because most changes will affect IVARQ data for more than one variable. For example, a change in dimension

---

[1]Make sure that the first IVARQ DATA card subscripts agree with IVARQ's dimension (256) in INP.

Table C-1. INPUT VARIABLE INFORMATION STORED IN INP

Storage Location	Contents	Example (for JV = 10)
IVARQ(JV,1)	Alphanumeric name of variable JV	6HCEANG
IVARQ(JV,2)	First dimension of variable JV.	14
IVARQ(JV,3)	Second dimension.	0
IVARQ(JV,4)	Third dimension.	0
IVARQ(JV,5)	Common location for first value of variable, compared to beginning of blank common (NEPD) at location 0.	269
IVARQ(JV,6)	Zero, for IDACASE.	0
IVARQ(JV,7)	Digit 1 gives variable type (1 = integer, 2 = real, 4 = alpha). Digit 2 is 0 for IDACASE [but not necessarily for MEDMOD].	20
IVARQ(JV,8)	All but last digit give common location for last variable value. Last digit gives number of dimensions.	2831

```
      COMMON   NEPD(1)
      COMMON   AENDBP(7,5),AHODA,ANGJMN(40),ANGJMX(40),ASMROL(100)
      COMMON   CBR1CI(25),CBR2CI(25),CDWNDS,CDWNH,CEANG(14),CERANG(14),
     1         CMAXCI(5),CMINCI(5),COFFUS,COFFH,CWAVE(3),COUR(100)
      COMMON   DETMIN(7),DISBP(7,5),DISSCL(7,10),DRLDSL,DRNGMN(3),
     1         DRNGMX(3),DSCA,DSDVEL,DSLKTS,DSSVEL,DTAFTR(40),
     2         DTBHSD(6),DTBSAL(40),DTDET,DTDLCI(5),DTDLSD(6),
     3         DTHOCI(6),DTILCI(5),DTILSD(6),DTLOAD(24),DTLOSD(6),
     4         DTOVRL(40),DTTRAN(40),DTTHGD(40)
      COMMON   ETRNUT(2),ETRXUT(2)
      COMMON   FAMITK,FLTHDG,FRMNOV(40)
      COMMON   GANGSD(6),GBR1SD(40),GBR2SD(40),G2MNSD(40),G2MXSD(40)
      COMMON   HDGA,HDGVEL,HDDA(100),HDDAD1,HDDAD2,HDDAD3,HDDVEL
      COMMON   IASMT(100),IATGTS(100),IDCKTS(40),IDEUFC,ILNCHR(40),
     1         INSALV,IPOUT,IPRSD,ISPRWS,ISHIPT(14),IZDWNR,IDTWAV,
     2         ITGTS(100),IZDNDS
      COMMON   JACT(40),JACTSD(6),JAMMO(24),JCONTH(40),JFRTSD(6),
     1         JLOCSD(40),JNLNSD(40),JSALVO(40),JSCCI(25),JSCSD(40),
     2         JSCSF(40),JTYPCI(25),JTYPSD(40)
      COMMON   MAXDET,MAXHIT,MAXKIL(14)
      COMMON   NASMSZ(5,3),NBP,NCASE,NDS,NIDNDS,NIDWNR,NIOFDS,NIOFFR,
     1         NMSLSD(6),NPRN,NRNDCI(5),NRNDSD(6),NS,NSCI,NSS,NSSD,
     2         NSYSCI(5),NSYSSD(6),NTRIAL,NUMWAV(5)
      COMMON   PKASM(40,3),PKCUM(10,2,5),PKFCMK(3),PKFMKR(4,3),
     1         PKS(3,14),PRUNPR(5),PRUNSZ(5),PTATDI(40)
      COMMON   RDETSK(7,10),RMAXSD(6,3),RMAXSK(40),RMINCI(5,2),
     1         RMINSD(6),RMINSK(40),RMNSKT,ROFFCI(5),RPKCUM(10,2,5),
     2         R1MND1(9,5,3),R1MND2(9,5,3),R1MND3(9,5,3),
     3         R1MXD1(9,5,3),R1MXD2(9,5,3),R1MXD3(9,5,3),
     4         R2MND1(9,5,3),R2MND2(9,5,3),R2MND3(9,5,3),
     5         R2MXD1(9,5,3),R2MXD2(9,5,3),R2MXD3(9,5,3)
      COMMON   SAMVEL(40),SECTMN(7,8),SECTMX(7,8),SEQPAR(2,2,2),
     1         SEQTES(2),SHANG1(14),SHANG2(14),SHANGT(14),SKVEL,
     2         SSPKSD(6,3)
      COMMON   TOCTIE(40),TOLINP(100),TOLPAR(3,2)
      COMMON   VGUNSD(6),VRNDCI(5)
```

(Continued)

Figure C-7. SAMPLE COMMON AND DATA STATEMENTS DESCRIBING INPUT VARIABLES

```
DIMENSION IVARQ(256,8)
DATA N/ 153/,(IVARQ(I,1),I= 154,256 )/ 103*6HZZZZZZ/
DATA(IVARQ(  1,K),K=1,8)/6HAENDBP,  7,  5, 0,    1,  0,20,  362/
DATA(IVARQ(  2,K),K=1,8)/6HAHOBA ,  1,  0, 0,   36,  0,20,  371/
DATA(IVARQ(  3,K),K=1,8)/6HANGJMN, 40,  0, 0,   37,  0,20,  771/
DATA(IVARQ(  4,K),K=1,8)/6HANGJMX, 40,  0, 0,   77,  0,20, 1171/
DATA(IVARQ(  5,K),K=1,8)/6HASMROL,100,  0, 0,  117,  0,20, 2171/
DATA(IVARQ(  6,K),K=1,8)/6HCBR1CI, 25,  0, 0,  217,  0,20, 2421/
DATA(IVARQ(  7,K),K=1,8)/6HCBR2CI, 25,  0, 0,  242,  0,20, 2671/
DATA(IVARQ(  8,K),K=1,8)/6HCDWNDS,  1,  0, 0,  267,  0,20, 2681/
DATA(IVARQ(  9,K),K=1,8)/6HCDWNR ,  1,  0, 0,  268,  0,20, 2691/
DATA(IVARQ( 10,K),K=1,8)/6HCEANG , 14,  0, 0,  269,  0,20, 2831/
DATA(IVARQ( 11,K),K=1,8)/6HCERANG, 14,  0, 0,  283,  0,20, 2971/
DATA(IVARQ( 12,K),K=1,8)/6HCMAXCI,  5,  0, 0,  297,  0,20, 3021/
DATA(IVARQ( 13,K),K=1,8)/6HCMINCI,  5,  0, 0,  302,  0,20, 3071/
DATA(IVARQ( 14,K),K=1,8)/6HCOFFDS,  1,  0, 0,  307,  0,20, 3081/
DATA(IVARQ( 15,K),K=1,8)/6HCOFFR ,  1,  0, 0,  308,  0,20, 3091/
DATA(IVARQ( 16,K),K=1,8)/6HCOUR  ,100,  0, 0,  312,  0,20, 4121/
DATA(IVARQ( 17,K),K=1,8)/6HCWAVE ,  3,  0, 0,  309,  0,20, 3121/
DATA(IVARQ( 18,K),K=1,8)/6HDETMIN,  7,  0, 0,  412,  0,20, 4191/
DATA(IVARQ( 19,K),K=1,8)/6HDISBP ,  7,  5, 0,  419,  0,20, 4542/
DATA(IVARQ( 20,K),K=1,8)/6HDISSGL,  7, 10, 0,  454,  0,20, 5242/
DATA(IVARQ( 21,K),K=1,8)/6HDRLDSL,  1,  0, 0,  524,  0,20, 5251/
DATA(IVARQ( 22,K),K=1,8)/6HDRNGMN,  3,  0, 0,  525,  0,20, 5281/
DATA(IVARQ( 23,K),K=1,8)/6HDRNGMX,  3,  0, 0,  528,  0,20, 5311/
DATA(IVARQ( 24,K),K=1,8)/6HDSCA  ,  1,  0, 0,  531,  0,20, 5321/
DATA(IVARQ( 25,K),K=1,8)/6HDSDVEL,  1,  0, 0,  532,  0,20, 5331/
DATA(IVARQ( 26,K),K=1,8)/6HDSLKTS,  1,  0, 0,  533,  0,20, 5341/
DATA(IVARQ( 27,K),K=1,8)/6HDSSVEL,  1,  0, 0,  534,  0,20, 5351/
DATA(IVARQ( 28,K),K=1,8)/6HDTAFTR, 40,  0, 0,  535,  0,20, 5751/
DATA(IVARQ( 29,K),K=1,8)/6HDTBRSD,  6,  0, 0,  575,  0,20, 5811/
DATA(IVARQ( 30,K),K=1,8)/6HDTBSAL, 40,  0, 0,  581,  0,20, 6211/
DATA(IVARQ( 31,K),K=1,8)/6HDTDET ,  1,  0, 0,  621,  0,20, 6221/
DATA(IVARQ( 32,K),K=1,8)/6HDTDLCI,  5,  0, 0,  622,  0,20, 6271/
DATA(IVARQ( 33,K),K=1,8)/6HDTDLSD,  6,  0, 0,  627,  0,20, 6331/
DATA(IVARQ( 34,K),K=1,8)/6HDTHOCI,  6,  0, 0,  633,  0,20, 6391/
DATA(IVARQ( 35,K),K=1,8)/6HDTILCI,  5,  0, 0,  639,  0,20, 6441/
DATA(IVARQ( 36,K),K=1,8)/6HDTILSD,  6,  0, 0,  644,  0,20, 6501/
DATA(IVARQ( 37,K),K=1,8)/6HDTLOAD, 24,  0, 0,  650,  0,20, 6741/
DATA(IVARQ( 38,K),K=1,8)/6HDTLOSD,  6,  0, 0,  674,  0,20, 6801/
DATA(IVARQ( 39,K),K=1,8)/6HDTOVRL, 40,  0, 0,  680,  0,20, 7201/
DATA(IVARQ( 40,K),K=1,8)/6HDTTRAN, 40,  0, 0,  720,  0,20, 7601/
DATA(IVARQ( 41,K),K=1,8)/6HDTTRGD, 40,  0, 0,  760,  0,20, 8001/
DATA(IVARQ( 42,K),K=1,8)/6HETRNUT,  2,  0, 0,  800,  0,20, 8021/
DATA(IVARQ( 43,K),K=1,8)/6HETRXUT,  2,  0, 0,  802,  0,20, 8041/
DATA(IVARQ( 44,K),K=1,8)/6HFAHTTK,  1,  0, 0,  804,  0,20, 8051/
```

Figure C-7. (Continued)

```
DATA(IVARQ( 45,K),K=1,8)/6HFLTHDG, 1, 0, 0, 805, 0,20, 8061/ 09MAY80
DATA(IVARQ( 46,K),K=1,8)/6HFRMNOV, 40, 0, 0, 806, 0,20, 8461/ 09MAY80
DATA(IVARQ( 47,K),K=1,8)/6HGANGSD, 6, 0, 0, 846, 0,20, 8521/ 09MAY80
DATA(IVARQ( 48,K),K=1,8)/6HGBR1SD, 40, 0, 0, 852, 0,20, 8921/ 09MAY80
DATA(IVARQ( 49,K),K=1,8)/6HGBR2SD, 40, 0, 0, 892, 0,20, 9321/ 09MAY80
DATA(IVARQ( 50,K),K=1,8)/6HG2MNSD, 40, 0, 0, 932, 0,20, 9721/ 09MAY80
DATA(IVARQ( 51,K),K=1,8)/6HG2MXSD, 40, 0, 0, 972, 0,20, 10121/ 09MAY80
DATA(IVARQ( 52,K),K=1,8)/6HHDCA  ,  1, 0, 0, 1012, 0,20, 10131/ 09MAY80
DATA(IVARQ( 53,K),K=1,8)/6HHDCVEL,  1, 0, 0, 1013, 0,20, 10141/ 09MAY80
DATA(IVARQ( 54,K),K=1,8)/6HHDDAD1,  1, 0, 0, 1114, 0,20, 11151/ 09MAY80
DATA(IVARQ( 55,K),K=1,8)/6HHDDAD2,  1, 0, 0, 1115, 0,20, 11161/ 09MAY80
DATA(IVARQ( 56,K),K=1,8)/6HHDDAD3,  1, 0, 0, 1116, 0,20, 11171/ 09MAY80
DATA(IVARQ( 57,K),K=1,8)/6HHDDA  ,100, 0, 0, 1014, 0,20, 11141/ 09MAY80
DATA(IVARQ( 58,K),K=1,8)/6HHDDVEL,  1, 0, 0, 1117, 0,20, 11181/ 09MAY80
DATA(IVARQ( 59,K),K=1,8)/6HIASMT ,100, 0, 0, 1118, 0,10, 12181/ 09MAY80
DATA(IVARQ( 60,K),K=1,8)/6HIATGTS,100, 0, 0, 1218, 0,10, 13181/ 09MAY80
DATA(IVARQ( 61,K),K=1,8)/6HIDCKTS, 40, 0, 0, 1318, 0,10, 13581/ 09MAY80
DATA(IVARQ( 62,K),K=1,8)/6HIDCUFC,  1, 0, 0, 1358, 0,10, 13591/ 09MAY80
DATA(IVARQ( 63,K),K=1,8)/6HIDTWAV,  1, 0, 0, 1418, 0,10, 14191/ 09MAY80
DATA(IVARQ( 64,K),K=1,8)/6HILNCHR, 40, 0, 0, 1359, 0,10, 13991/ 09MAY80
DATA(IVARQ( 65,K),K=1,8)/6HINSALV,  1, 0, 0, 1399, 0,10, 14001/ 09MAY80
DATA(IVARQ( 66,K),K=1,8)/6HIPOUT ,  1, 0, 0, 1400, 0,10, 14011/ 09MAY80
DATA(IVARQ( 67,K),K=1,8)/6HIPRSD ,  1, 0, 0, 1401, 0,10, 14021/ 09MAY80
DATA(IVARQ( 68,K),K=1,8)/6HISFRHS,  1, 0, 0, 1402, 0,10, 14031/ 09MAY80
DATA(IVARQ( 69,K),K=1,8)/6HISHIPT, 14, 0, 0, 1403, 0,10, 14171/ 09MAY80
DATA(IVARQ( 70,K),K=1,8)/6HITGTS ,100, 0, 0, 1419, 0,10, 15191/ 09MAY80
DATA(IVARQ( 71,K),K=1,8)/6HIZDNDS,  1, 0, 0, 1519, 0,10, 15201/ 09MAY80
DATA(IVARQ( 72,K),K=1,8)/6HIZDWNR,  1, 0, 0, 1417, 0,10, 14181/ 09MAY80
DATA(IVARQ( 73,K),K=1,8)/6HJACTSD,  6, 0, 0, 1560, 0,10, 15661/ 09MAY80
DATA(IVARQ( 74,K),K=1,8)/6HJACT  , 40, 0, 0, 1520, 0,10, 15601/ 09MAY80
DATA(IVARQ( 75,K),K=1,8)/6HJAMMO , 24, 0, 0, 1566, 0,10, 15901/ 09MAY80
DATA(IVARQ( 76,K),K=1,8)/6HJCONTR, 40, 0, 0, 1590, 0,10, 16301/ 09MAY80
DATA(IVARQ( 77,K),K=1,8)/6HJFRTSD,  6, 0, 0, 1630, 0,10, 16361/ 09MAY80
DATA(IVARQ( 78,K),K=1,8)/6HJLOCSD, 40, 0, 0, 1636, 0,10, 16761/ 09MAY80
DATA(IVARQ( 79,K),K=1,8)/6HJNLNSD, 40, 0, 0, 1676, 0,10, 17161/ 09MAY80
DATA(IVARQ( 80,K),K=1,8)/6HJSALVO, 40, 0, 0, 1716, 0,10, 17561/ 09MAY80
DATA(IVARQ( 81,K),K=1,8)/6HJSCCI , 25, 0, 0, 1756, 0,10, 17811/ 09MAY80
DATA(IVARQ( 82,K),K=1,8)/6HJSCSD , 40, 0, 0, 1781, 0,10, 18211/ 09MAY80
DATA(IVARQ( 83,K),K=1,8)/6HJSCSF , 40, 0, 0, 1821, 0,10, 18611/ 09MAY80
DATA(IVARQ( 84,K),K=1,8)/6HJTYPCI, 25, 0, 0, 1861, 0,10, 18861/ 09MAY80
DATA(IVARQ( 85,K),K=1,8)/6HJTYPSD, 40, 0, 0, 1886, 0,10, 19261/ 09MAY80
DATA(IVARQ( 86,K),K=1,8)/6HMAXDET,  1, 0, 0, 1926, 0,10, 19271/ 09MAY80
DATA(IVARQ( 87,K),K=1,8)/6HMAXHIT,  1, 0, 0, 1927, 0,10, 19281/ 09MAY80
DATA(IVARQ( 88,K),K=1,8)/6HMAXKIL, 14, 0, 0, 1928, 0,10, 19421/ 09MAY80
DATA(IVARQ( 89,K),K=1,8)/6HNASMSZ,  5, 3, 0, 1942, 0,10, 19572/ 09MAY80
DATA(IVARQ( 90,K),K=1,8)/6HNBP   ,  1, 0, 0, 1957, 0,10, 19581/ 09MAY80
DATA(IVARQ( 91,K),K=1,8)/6HNCASE ,  1, 0, 0, 1958, 0,10, 19591/ 09MAY80
DATA(IVARQ( 92,K),K=1,8)/6HNDS   ,  1, 0, 0, 1959, 0,10, 19601/ 09MAY80
DATA(IVARQ( 93,K),K=1,8)/6HNEPD  ,  1, 0, 0,    0, 0,10,    11/ 09MAY80
DATA(IVARQ( 94,K),K=1,8)/6HNIDNDS,  1, 0, 0, 1960, 0,10, 19611/ 09MAY80
DATA(IVARQ( 95,K),K=1,8)/6HNIDWNR,  1, 0, 0, 1961, 0,10, 19621/ 09MAY80
DATA(IVARQ( 96,K),K=1,8)/6HNIOFDS,  1, 0, 0, 1962, 0,10, 19631/ 09MAY80
DATA(IVARQ( 97,K),K=1,8)/6HNIOFFR,  1, 0, 0, 1963, 0,10, 19641/ 09MAY80
DATA(IVARQ( 98,K),K=1,8)/6HNMSLSD,  6, 0, 0, 1964, 0,10, 19701/ 09MAY80
DATA(IVARQ( 99,K),K=1,8)/6HNPRN  ,  1, 0, 0, 1970, 0,10, 19711/ 09MAY80
DATA(IVARQ(100,K),K=1,8)/6HNRNDCI,  5, 0, 0, 1971, 0,10, 19761/ 09MAY80
DATA(IVARQ(101,K),K=1,8)/6HNRNDSD,  6, 0, 0, 1976, 0,10, 19821/ 09MAY80
```

Figure C-7. (Continued)

```
DATA(IVARQ(102,K),K=1,8)/6HNSCI    ,  1,  0, 0, 1983,   0,10, 19841/ 09MAY80
DATA(IVARQ(103,K),K=1,8)/6HNSSD    ,  1,  0, 0, 1985,   0,10, 19861/ 09MAY80
DATA(IVARQ(104,K),K=1,8)/6HNSS     ,  1,  0, 0, 1984,   0,10, 19851/ 09MAY80
DATA(IVARQ(105,K),K=1,8)/6HNSYSCI,  5,  0, 0, 1986,   0,40, 19911/ 09MAY80
DATA(IVARQ(106,K),K=1,8)/6HNSYSSD,  6,  0, 0, 1991,   0,40, 19971/ 09MAY80
DATA(IVARQ(107,K),K=1,8)/6HNS      ,  1,  0, 0, 1982,   0,10, 19831/ 09MAY80
DATA(IVARQ(108,K),K=1,8)/6HNTRIAL,  1,  0, 0, 1997,   0,10, 19981/ 09MAY80
DATA(IVARQ(109,K),K=1,8)/6HNUMWAV,  5,  0, 0, 1998,   0,10, 20031/ 09MAY80
DATA(IVARQ(110,K),K=1,8)/6HPKASM  , 40,  3, 0, 2003,   0,20, 21232/ 09MAY80
DATA(IVARQ(111,K),K=1,8)/6HPKCUM  , 10,  2, 5, 2123,   0,20, 22233/ 09MAY80
DATA(IVARQ(112,K),K=1,8)/6HPKFCMK,  3,  0, 0, 2223,   0,20, 22261/ 09MAY80
DATA(IVARQ(113,K),K=1,8)/6HPKFMKR,  4,  3, 0, 2226,   0,20, 22382/ 09MAY80
DATA(IVARQ(114,K),K=1,8)/6HPKS     ,  3, 14, 0, 2238,   0,20, 22802/ 09MAY80
DATA(IVARQ(115,K),K=1,8)/6HPRUNPR,  5,  0, 0, 2280,   0,20, 22851/ 09MAY80
DATA(IVARQ(116,K),K=1,8)/6HPRUNSZ,  5,  0, 0, 2285,   0,20, 22901/ 09MAY80
DATA(IVARQ(117,K),K=1,8)/6HPTATOI, 40,  0, 0, 2290,   0,20, 23301/ 09MAY80
DATA(IVARQ(118,K),K=1,8)/6HRDETSK,  7, 10, 0, 2330,   0,20, 24002/ 09MAY80
DATA(IVARQ(119,K),K=1,8)/6HRMAXSD,  6,  3, 0, 2400,   0,20, 24182/ 09MAY80
DATA(IVARQ(120,K),K=1,8)/6HRMAXSK, 40,  0, 0, 2418,   0,20, 24581/ 09MAY80
DATA(IVARQ(121,K),K=1,8)/6HRMINCI,  5,  2, 0, 2458,   0,20, 24682/ 09MAY80
DATA(IVARQ(122,K),K=1,8)/6HRMINSD,  6,  0, 0, 2468,   0,20, 24741/ 09MAY80
DATA(IVARQ(123,K),K=1,8)/6HRMINSK, 40,  0, 0, 2474,   0,20, 25141/ 09MAY80
DATA(IVARQ(124,K),K=1,8)/6HRMNSKT,  1,  0, 0, 2514,   0,20, 25151/ 09MAY80
DATA(IVARQ(125,K),K=1,8)/6HROFFCI,  5,  0, 0, 2515,   0,20, 25201/ 09MAY80
DATA(IVARQ(126,K),K=1,8)/6HRPKCUM, 10,  2, 5, 2520,   0,20, 26203/ 09MAY80
DATA(IVARQ(127,K),K=1,8)/6HR1MND1,  9,  5, 3, 2620,   0,20, 27553/ 09MAY80
DATA(IVARQ(128,K),K=1,8)/6HR1MND2,  9,  5, 3, 2755,   0,20, 28903/ 09MAY80
DATA(IVARQ(129,K),K=1,8)/6HR1MND3,  9,  5, 3, 2890,   0,20, 30253/ 09MAY80
DATA(IVARQ(130,K),K=1,8)/6HR1MXD1,  9,  5, 3, 3025,   0,20, 31603/ 09MAY80
DATA(IVARQ(131,K),K=1,8)/6HR1MXD2,  9,  5, 3, 3160,   0,20, 32953/ 09MAY80
DATA(IVARQ(132,K),K=1,8)/6HR1MXD3,  9,  5, 3, 3295,   0,20, 34303/ 09MAY80
DATA(IVARQ(133,K),K=1,8)/6HR2MND1,  9,  5, 3, 3430,   0,20, 35653/ 09MAY80
DATA(IVARQ(134,K),K=1,8)/6HR2MND2,  9,  5, 3, 3565,   0,20, 37003/ 09MAY80
DATA(IVARQ(135,K),K=1,8)/6HR2MND3,  9,  5, 3, 3700,   0,20, 38353/ 09MAY80
DATA(IVARQ(136,K),K=1,8)/6HR2MXD1,  9,  5, 3, 3835,   0,20, 39703/ 09MAY80
DATA(IVARQ(137,K),K=1,8)/6HR2MXD2,  9,  5, 3, 3970,   0,20, 41053/ 09MAY80
DATA(IVARQ(138,K),K=1,8)/6HR2MXD3,  9,  5, 3, 4105,   0,20, 42403/ 09MAY80
DATA(IVARQ(139,K),K=1,8)/6HSAMVEL, 40,  0, 0, 4240,   0,20, 42801/ 09MAY80
DATA(IVARQ(140,K),K=1,8)/6HSECTMN,  7,  8, 0, 4280,   0,20, 43362/ 09MAY80
DATA(IVARQ(141,K),K=1,8)/6HSECTMX,  7,  8, 0, 4336,   0,20, 43922/ 09MAY80
DATA(IVARQ(142,K),K=1,8)/6HSEQPAR,  2,  4, 2, 4392,   0,20, 44003/ 09MAY80
DATA(IVARQ(143,K),K=1,8)/6HSEQTES,  2,  0, 0, 4400,   0,20, 44021/ 09MAY80
DATA(IVARQ(144,K),K=1,8)/6HSHANG1, 14,  0, 0, 4402,   0,20, 44161/ 09MAY80
DATA(IVARQ(145,K),K=1,8)/6HSHANG2, 14,  0, 0, 4416,   0,20, 44301/ 09MAY80
DATA(IVARQ(146,K),K=1,8)/6HSHAWGT, 14,  0, 0, 4430,   0,20, 44441/ 09MAY80
DATA(IVARQ(147,K),K=1,8)/6HSKVEL  ,  1,  0, 0, 4444,   0,20, 44451/ 09MAY80
DATA(IVARQ(148,K),K=1,8)/6HSSPKSD,  6,  3, 0, 4445,   0,20, 44632/ 09MAY80
DATA(IVARQ(149,K),K=1,8)/6HTOCTIE, 40,  0, 0, 4463,   0,20, 45031/ 09MAY80
DATA(IVARQ(150,K),K=1,8)/6HTOLINP,100,  0, 0, 4503,   0,20, 46031/ 09MAY80
DATA(IVARQ(151,K),K=1,8)/6HTOLPAR,  3,  2, 0, 4603,   0,20, 46092/ 09MAY80
DATA(IVARQ(152,K),K=1,8)/6HVGUNSD,  6,  0, 0, 4609,   0,20, 46151/ 09MAY80
DATA(IVARQ(153,K),K=1,8)/6HVRNOCI,  5,  0, 0, 4615,   0,20, 46201/ 09MAY80
```

Figure C-7. (Concluded)

for one variable changes storage locations for all variables following in common.

Producing and punching the information for the IVARQ DATA cards is possible, but tedious. A program COMM exists which takes the cards of a blank common block and produces the corresponding set of DATA statements. The changed COMMON cards and new DATA cards then can be introduced to IDACASE (in a single update run for IDA's program library).

There exists at IDA a more automated procedure for accomplishing all these steps with a single computer run starting with the changed COMMON cards. The procedure utilizes program COMM and a variety of updating procedures.

3. Incompatibilities for Machine Conversion in INP

The INP subroutine, as it exists in IDACASE, uses capabilities of the CDC 6400 not available on many other computers, although this is a FORTRAN subroutine. The main problems for conversion of INP to other computers are:

1. The ENCODE/DECODE statements for memory transfer of data,
2. The use of larger CDC word size (60 bit words containing up to 10 characters),
3. Character to integer conversion (probably not a concern for IDACASE as more than 99 cases are required before conversion becomes necessary).

A version of the INP subroutine is operational on a Honeywell 6000 series computer, indicating the possibility of conversion and existence of a non-CDC version of this routine. It should be noted that INP only reads inputs; it does not affect the calculations made by the model. Therefore, a potential user of IDACASE who does not want to convert INP to another computer can replace INP with an input routine appropriate for the user's computer.

APPENDIX B

TABULAR GUIDES TO THE MEDMOD COMPUTER PROGRAM

CONTENTS OF APPENDIX B

TABLES

APPENDIX B

TABULAR GUIDES TO THE MEDMOD COMPUTER PROGRAM

This appendix consists of two parts. The first part introduces (and explains, where necessary) the tables presented in the second part. The reason for grouping the tables together in the second part (rather than interspersing the tables and the discussions concerning them) is to make it easier to refer to these tables when examining the MEDMOD computer program. That is, these tables are primarily designed to be referenced as guides to the MEDMOD computer program, not just to be read at one sitting before looking at that program.

A. INTRODUCTION TO THE TABLES

Table B-1 gives an exhaustive list of all program segments (overlays, subroutines, and functions) defined in the MEDMOD computer program in the order in which they appear in that program. This table not only summarizes information that can be found by looking at each segment individually; it also lists, for each program segment, all the other program segments that call that segment.

Since the segments are listed in Table B-1 in the same order as they appear in the code, one can also use Table B-1 to find the location of each program segment in the computer code. To assist in doing this, Table B-2 lists the program segments of MEDMOD in alphabetical order and gives the segment number (i.e., the position in Table B-1) of each program segment.

Table B-3 lists each labeled COMMON block, it lists the program segments in which each of these labeled COMMON blocks appear, and it lists and (perhaps most important) defines each variable that appears in each of these labeled COMMON blocks.

Table B-4 gives an exhaustive list of those inputs that are resource variables in MEDMOD. Index limits and definitions of these inputs are also given. These inputs (and only these inputs) are incremental inputs in Subroutine TIMET.

Table B-5 displays which major subroutines of MEDMOD use which of these resource variables. Note that all resource variables are inputs, all inputs are in blank COMMON, and only overlays and major subroutines of MEDMOD contain the blank COMMON block. Thus, no subroutines other than those listed in Table B-5 directly reference these resource variables.

Table B-6 defines and gives other relevant information concerning the major indexing variables used in MEDMOD.

Among other things, Table B-7 states which program segments use which limit variables in MEDMOD. Tables 1 through 11 and 13 of Chapter II list all the inputs except limit inputs that are used by each major subroutine; thus, Table B-7, which gives the limit inputs used by each major subroutine, completes the categorization of inputs to the major subroutines of MEDMOD using them.

Most of the input parameters to MEDMOD are used in exactly one of its major subroutines. Table B-8 lists those input parameters that are used in more than one major subroutine. Thus, special care should be taken in preparing data for, or altering the code involving, any of the parameters listed in Table B-8.

MEDMOD contains no automated internal data-checking routines. Tables B-7 (for limit variables) and B-9 (for other variables) give some computer code-related restrictions on input variables. Of course, there are many logical restrictions (such as all input probabilities should be between zero and one, inclusive) in addition to the code-related restrictions listed in these tables. Table B-9 is in three parts. Part 1 presents input variables which should be strictly greater than zero. (No input variable should ever be less than zero.) Part 1 does not include the limit variables listed in Table B-7. Part 2 lists integer input variables that can only take on a few values without causing execution errors or meaningless output. Most of these variables index "kind of protocol" or "kind of equation." Part 3 lists some input variables whose values must satisfy more complicated restrictions.

Reference [1] assigns an order number to most of the inputs to its model, and it defines and discusses these inputs in order by their number (these numbers run from 1 to 88 for the 88 inputs that are assigned numbers in [1]--but not all of these 88 inputs are used in MEDMOD). All inputs except for "task force components" (i.e., Blue resources) are assigned numbers in [1]. To assist in using the data and discussions of [1], Table B-11 lists each input used in Subroutine CTFMOD (in alphabetical order) in its first column; if there is a corresponding variable in the R-245 model, it lists the R-245 name of that corresponding variable in the second column and the order number of that variable (if it has one) or the words "Blue Resource" (if it doesn't) in the third column. With one exception, if there is no R-245 input that corresponds to the input in the first column, then "none" is listed in the second column and the third column is blank. The one exception is the input NKRB, which is logically fixed at 1 in the R-245 model.

Table B-12 lists, in alphabetical order, every input variable, every variable in labeled COMMON (all inputs and only inputs are in blank COMMON), every major indexing variable, and every computed limit variable used in MEDMOD. The first column of that table gives the name of the variable. The second column gives the type and (where appropriate) the subtype of the corresponding variable. All variables are categorized into exactly one of the following types: input, labeled COMMON, indexing, or computed limit. Subtypes of input variables are: resource, parameter, or limit. The names of the labeled COMMON block comprise the subtypes for labeled common variables. There are no subtypes for indexing or computed limit variables. The third column lists all program segments using this variable.

The potential utility of the third column of Table B-12 should be clear. However, the second column of this table is also quite useful because it serves as an index to the definitions of variables contained in Tables B-3, B-6, B-7, and in Appendix C. To find the definitions of any variables listed in Table B-12, one should proceed as follows. All input variables are defined (in alphabetical order) in Appendix C. (Additional information concerning resource inputs is given in Table B-4, and additional information concerning limit input is given in Table B-7.) All variables in labeled common blocks are defined in Table B-3. Indexing variables are defined in Table B-6, and computed limits are defined in Table B-7.

B. TABLES

Tables B-1 through B-12, which follow, conclude this appendix.

Table B-1. Program Segments of MEDMOD and Their Characteristics

	Name of Program Segment	Mnemonic	Purpose of This Program Segment	Parameter List[i]	This Program Segment is Called by	Program Segments Called by This Program Segment	COMMON Blocks Appearing in This Program Segment
1.	DRIVER [a]	Driving Program	Calls INP and MEDMOD; prints headings for summary table	(n/a)	beginning of program	INP, MEDMOD	blank COMMON, COMIGO COMOUT
2.	MEDMOD [a]	Mediterranean Model	Control program for the combat simulation	(n/a)	DRIVER	Program Segment names noted by c in Column 2	blank COMMON, COMCTF, COMGA, COMIGO, COMSOR COMOUT
3.	LOCTFF [b]	Locate Task Force Function	Determines what region the task force is in, for each time period ITP.	ITP, LGTHMP, LTFMP, MIMP	MOVTF	(none)	(none)
4.	ABATCK [c]	Airbase Attack	Models Blue air attack on vulnerable Red airbase	L	MEDMOD	AIRAIR, ATRTSS, ATRTAB	blank COMMON, COMCTF, COMGA, COMSOR
5.	ADDMOE [c]	Add up measures of effectiveness	Determines whether to stop the simulation	ITP, ISTOP	MEDMOD	(none)	blank COMMON, COMCTF
6.	AIRAIR [d]	Air vs. Air	Computes Air-to-Air attrition for escorts vs. defenders and then for defenders vs. attackers	g	ABATCK PLBAB	ATRTED, ATRTDA	(none)
7.	ATRTAB [d]	Attrition at Airbase	Computes attrition to Red aircraft on vulnerable Red airbase	g	ABATCK	(none)	(none)
8.	ATRTDA [d]	Attrition: Defenders vs. Attackers	Computes Air-to-Air attrition: Defenders vs. Attackers	g	AIRAIR	BINFAC	(none)
9.	ATRTED [d]	Attrition: Escorts vs. Defenders	Computes Air-to-Air attrition: Escort aircraft vs. Defenders	g	AIRAIR	BINOAT	(none)
10.	ATRTIA [d]	Attrition: Interceptors vs. Attackers	Computes attrition: Blue CAP and DLI aircraft vs. Red bomber and escort aircraft	g	CTFMOD	(none)	(none)
11.	ATRTSS [d]	Attrition by SAMs	Computes attrition for aircraft vs. SAMs	g	ABATCK POWERP	BINOAT	(none)
12.	BARKCK [d]	Barrier Kills and Counter Kills	Assesses kills by barrier submarines against enemy penetrators; also counter-kills	g	MOVRS MOVTF	(none)	BARSCK
13.	BINFAC [d]	Binomial Attrition Factor	Binomial attrition routine; computes a fraction of targets not killed	g	ATRTDA	(none)	(none)
14.	BINOAT [d]	Binomial Attrition	Binomial attrition routine (Heterogeneous Lanchester linear analog)	g	ATRTED ATRTSS SUBSUB	(none)	(none)
15.	BINOM [b]	Binomial (Distribution Probabilities)	Computes Probability of M successes in N trials when prob. of success on a trial is P.	N,M,P	MOVTF	(none)	(none)
16.	CTFMOD [c]	Carrier Task Force Model	Exercises the Carrier Task Force Model Based on IDA Report R-245.	L	MEDMOD	ATRTIA, FUNCT1, FUNCT2, FUNCT3, FUNCT5, FUNCT6, FUNCT9, FUNC11, FUNC12	blank COMMON, COMCTF, COMGA, COMSOR
17.	FUNCT1 [b]	Function - 1	Computes a quantity necessary for CTFMOD	X, T1, T2, T3, T4	CTFMOD	(none)	(none)
18.	FUNCT2 [b]	Function - 2	" "	X, AEWD, STAR, THSECA	CTFMOD	(none)	(none)
19.	FUNCT3 [b]	Function - 3	" "	X, CAPSTR, D1	CTFMOD	(none)	(none)
20.	FUNCT5 [b]	Function - 5	" "	X	CTFMOD	(none)	(none)

Notes on next page.

(Continued)

Table B-1. (Concluded)

	Name of Program Segment	Mnemonic	Purpose of This Program Segment	Parameter List[i]	This Program Segment is Called By	Program Segments Called By This Program Segment	COMMON Blocks Appearing in This Program Segment
21.	FUNCT6 [b]	Function - 6	Computes a quantity necessary for CTFMOD	X, ESLR, ESR, SUBSOR	CTFMOD	(none)	(none)
22.	FUNCT9 [b]	Function - 9	" "	Y, TAB1Ø	CTFMOD	FUNC1Ø	(none)
23.	FUNC10 [b]	Function - 10	" "	X, TAB1Ø	FUNCT9	(none)	(none)
24.	FUNC11 [b]	Function - 11	" "	X, FPPL2	CTFMOD	(none	(none)
25.	FUNC12 [b]	Function - 12	Computes probability of carrier destruction as a function of torpedo hits sustained by carrier	X, TAB12	CTFMOD MOVTF	(none)	(none)
26.	DDAY [c]	D-Day Shoot Out	Models the D-Day Shoot out	L	MEDMOD	(none)	blank COMMON, COMCTF
27.	GNAATK [c]	Generate Air Attack	Generates Red air attacks on the Task Force	L, ITP	MEDMOD	(none)	blank COMMON, COMGA
28.	MOVRS [c]	Move Red Ships	Moves Red ships (incl. submarines) from region to region, assessing barrier attrition as appropriate	LOCTF, ITP	MEDMOD	BARKCK	blank COMMON
29.	MOVTF [c]	Move Task Force	Moves the (Blue) Task Force from region to region as appropriate, assessing barrier attrition as necessary	LOCTF, ITP	MEDMOD	LOCTFF, BARKCK, BINOM, FUNC12	blank COMMON, BARSCK, COMCTF
30.	PLBAB [c]	Penetrate Land-Based Air Barrier	Models the attempt by the Red air attack to penetrate the Blue land-based air barrier	L	MEDMOD	AIRAIR	blank COMMON, COMGA
31.	POWERP [c]	Power Projection	Calculates power projection results	L, ITP	MEDMOD	ATRTSS	blank COMMON, COMCTF, COMSOR, COMOUT
32.	PRTRES [f]	Print Resources	(This routine is not coded)		MEDMOD	(none)	(none)
33.	PRTSUM [c]	Print Summary Information	Every time period computes (& writes on tape 10) a line of information for the summary printout	LC, ITP	MEDMOD	(none)	blank COMMON, COMCTF, COMOUT
34.	SHPSHP [c]	Ships vs. Ships	Models Surface Ship vs. Surface Ship warfare; also, aircraft from blue carrier killing Red surface ships	L, ITP	MEDMOD	(none)	blank COMMON, COMCTF, COMSOR
35.	SUBSUB [c]	Submarines vs. Submarines	Models Blue Sub/Red Sub and Blue Sub/Red Surface Ship Interactions	L	MEDMOD	BINOAT	blank COMMON
36.	TIMET [c]	Time T (update inputs)	Changes or increments input variables when desired	ICYCLE (equivalent to ITP)	MEDMOD	EOF [e]	blank COMMON, COMIGO
37.	INP [a]	Inputs	Reads and prints out input variables	(n/a)	DRIVER	(none)	blank COMMON, COMIGO

NOTES

[a]Main program, overlaid

[b]Function subprogram

[c]Subroutine called by Program MEDMOD ("major" subroutine)

[d]Subroutine called by a program segment other than Program MEDMOD

[e]EOF is an internal function signifying end of file (Tape 15 here). May require attention in conversion to other machines.

[f]This subroutine is not currently coded.

[g]Parameter list for this subroutine is long; see the code.

[h]Program segments are listed in the order they appear in the MEDMOD code. Subroutines appear in alphabetical order; function subprograms appear after the program segment with which they are most closely associated. See Table B-2 for alphabetical list.

[i]L, LC, or LOCTF is the location (region) the task force is in; ITP is the current time period.

Table B-2. Program Segments of MEDMOD Listed in Alphabetical Order

Segment	Segment Number[a]	Segment	Segment Number[a]
ABATCK	4	FUNCT5	20
ADDMOE	5	FUNCT6	21
AIRAIR	6	FUNCT9	22
ATRTAB	7	FUNC10	23
ATRTDA	8	FUNC11	24
ATRTED	9	FUNC12	25
ATRTIA	10	GNAATK	27
ATRTSS	11	INP	37
BARKCK	12	LOCTFF	3
BINFAC	13	MEDMOD	2
BINOAT	14	MOVRS	28
BINOM	15	MOVTF	29
CTFMOD	16	PLBAB	30
DDAY	26	POWERP	31
DRIVER	1	PRTRES	32
FUNCT1	17	PRTSUM	33
FUNCT2	18	SHPSHP	34
FUNCT3	19	SUBSUB	35
		TIMET	36

[a]See Table B-1, _supra_.

Table B-3. MEDMOD COMMON Blocks and Their Properties

Blank COMMON contains all the input variables and no other variables. It appears in routines DRIVER, MEDMOD, ABATCK, ADDMOE, CTFMOD, DDAY, GNAATK, MOVRS, MOVTF, PLBAB, POWERP, PRTSUM, SHPSHP, SUBSUB, TIMET, and INP.

Labeled COMMON blocks are listed below:

Labeled COMMON Block	Program segments in which this block appears	Variables in this block (in order of appearance) and their definitions
BARSCK	BARKCK, MOVTF	in BARKCK / in MOVTF SIBCK1 / SCK31 — Number of barrier submarines counterkilled when barrier submarines shoot first. SIBCK2 / SCK32 — Number of barrier submarines counterkilled when penetrating ships shoot first.
COMCTF	MEDMOD, ABATCK, ADDMOE, CTFMOD, DDAY, MOVTF, POWERP, PRTSUM, SHPSHP	XEFFCM - Relative carrier effectiveness. (Intial value is 1.0, value decreases as carriers suffer successful attacks.) FGHTRI - Initial number of Blue fighter aircraft in the task force. ATTCKI - Initial number of Blue attack aircraft in the task force. XCAPST - Current number of CAP stations which are desired (as determined by the number of AEW aircraft and related parameters) and can be supported (as determined by the number of available fighters on the carriers).
COMGA	MEDMOD. ABATCK, CTFMOD, GNAATK, PLBAB	NTPSLA - Number of time periods that have elapsed since the last Red air attack on the task force. BMR(2,3) - BMR(I,K): Number of Red Bombers of Type K from Airbase I that are currently alive and are continuing in the air attack on the task force. ESC(2) - ESC(I): Number of Red fighters from Airbase I that are currently alive and are continuing to escort bombers in the air attack on the task force.
COMIGO	DRIVER, MEDMOD, TIMET, INP	IGO - Indicator for whether any input variables are to be changed in current time period.
COMOUT	DRIVER, MEDMOD, POWERP, PRTSUM	CWPPAS - Cumulative weighted power projection sorties successfully flown. CWTPTF - Cumulative weighted effectiveness of the task force. PPSORT - Number of power projection sorties successfully flown during the time period. NTPSIM - Number of time periods actually simulated. LTASKF(90) - LTASKF(ITP): Location of the task force in time period ITP.
COMSOR	MEDMOD, ABATCK, CTFMOD, POWERP, SHPSHP	FTSORU - Number of fighter aircraft whose sorties have been "used up" during the clock time period. ATSORU - Number of attack aircraft whose sorties have "been used up" during the clock time period.

Table B-4. Resource Variables by Side and Type

	VARIABLE NAME, INDICES[b] and LIMITS[c]	DEFINITION
BLUE RESOURCES		
Submarines	1. BSSNDS	Direct support submarines[d]
	2. BSIBAR(IBAR), IBAR=1,NLOC[e]	Submarines in barriers, by barrier
Surface Ships	3. XPLAT (with effectiveness XEFFCM)	Aircraft carriers
	4. XEAAW	Anti-air warfare escort ships
	5. XEASWA	Air-capable anti-submarine warfare escort ships
	6. XEASWN	Non-air-capable anti-submarine warfare escort ships
	7. XURGS	URG ships
Aircraft	8. XATTCK	Attack aircraft (total over all carriers)
	9. XFGHTR	Fighter aircraft (total over all carriers)
	10. XAEW	Carrier-based AEW aircraft
	11. XAEWLQ(L), L=1,NLOC	Land-based AEW aircraft available when task force is in region L.
	12. XASW	Carrier-based ASW aircraft
	13. XASWLQ(L), L=1,NLOC	Land-based ASW aircraft available when task force in in region L.
	14. PLBLBD(KBD,LB) KBD=1,NKBDPL LB =1,NLOC	Land-based air barrier aircraft, by kind and region. (Each region has an associated land base LB)
RED RESOURCES		
Submarines	15. RS(1,L), L=1,NLOC	Torpedo-firing submarines, by region
	16. RS(2,L), L=1,NLOC	Missile-firing submarines, by region
	17. RSIBAR(IBAR), IBAR=1, NLOC	Submarines in barriers, by barrier
Surface Ships	18. RS(KRS,L) KRS=3,NKRS L =1,NLOC	Red surface ships, by kind and region
Aircraft	19. ATABT(IAB,KRB) KRB=1,NKRB IAB=1,2	Bombers, by kind of bomber and airbase IAB = 1--vulnerable Red airbase 2--invulnerable Red airbase
	20. AESCAB(IAB), IAB=1,2	Escort aircraft, by airbase
	21. AINTCT	Interceptor aircraft (on vulnerable Red airbase only)
	22. SHEL	Aircraft shelters (on vulnerable Red airbase only)
SAMs	23. ABANM(KRSAM[f]), KRSAM=1,NABSAM	Actual number of missiles for SAMs defending the vulnerable Red airbase.
	24. ABRSAM(KRSAM), KRSAM=1,NABSAM	SAMs defending the vulnerable Red airbase, by kind
	25. PPANMS(KRS[f]), KRS=1,NPPSAM	Actual number of missiles for SAMs defending against Blue power projection
	26. PPRSAM(KRS), KRS=1,NPPSAM	SAMs defending against Blue power projection, by kind of SAM

[a]The initial amount of each resource is input; the variables are updated appropriately as the interactions of the simulation occur. The 26 entries above comprise 24 distinct inputs, since RS appears on 3 entries. These 24 inputs (and only these inputs) are considered as incremental inputs by Subroutine TIMET.

[b]See Table B-6, infra, for definitions of indexing variables used here.

[c]See Table B-7, infra, for definitions of limit variables used here.

[d]I.e., variable BSSNDS is to be interpreted as the number of Blue direct support submarines; similarly for the other resource variables.

[e]"IBAR=1,NLOC" means that IBAR varies from 1 through NLOC (inclusive); similarly for other indices..

[f]Variable KRSAM indexes Red SAMs in Subroutine ABATCK; variable KRS indexes Red SAMs in Subroutine POWERP.

Table B-5. Resource Variables (in Alphabetical Order) and Major Subroutines Using Them[a]

Resource Variable	Variable Number[b]	ABATCK	ADDMOE	CTFMOD	DDAY	GNAATK	MOVRS	MOVTF	PLBAB	POWERP	PRTSUM	SHPSHP	SUBSUB
ABANM(KRSAM)	23	*											
ABRSAM(KRSAM)	24	*											
AESCAB(IAB)	20	*		*		*			*		*		
AINTCT	21	*									*		
ATABT(IAB,KRB)	19	*		*		*			*		*		
BSIBAR(IBAR)	2						*				*		
BSSNDS	1							*			*		*
PLBLBD(KBD,LB)	14								*		*		
PPANMS(KRSAM)[c]	25									*			
PPRSAM(KRSAM)[c]	26									*			
RS(1,L)	15			*	*		*				*		*
RS(2,L)	16			*	*		*				*		*
RS(KRS,L), KRS ≥ 3	18				*		*				*	*	*
RSIBAR(IBAR)	17							*			*		
SHEL	22	*											
XAEW	10			*									
XAEWLQ(L)	11			*									
XASW	12			*									
XASWLQ(L)	13			*									
XATTCK	8	*		*	*					*	*	*	
XEAAW	4		*	*	*			*			*	*	
AEASWA	5		*	*	*			*			*	*	
XEASWN	6		*	*	*			*			*	*	
XFGHTR	9	*		*	*					*	*	*	
XPLAT (with XEFFCM)	3		*	*	*			*			*	*	
XURGS	7		*	*	*			*			*	*	

a * denotes that the resource variable is used in the indicated subroutine.

b See Table B-4, supra.

c The indexing variable actually used in the POWERP code is KRS. For clarity, KRSAM is used here.

Table B-6. Indexing Variables

NOTE: This table defines the most commonly used variables that index resource variables and parameters in the major MEDMOD subroutines. Same indexing variables have different meanings in different subroutines; all are given below. Variables are listed in alphabetical order.

INDEXING VARIABLE	PROGRAM SEGMENT(S) WHERE USED	VARIES From	VARIES To[a]	DEFINITION
I	ABATCK	1	2	Kind of Blue aircraft attacking vulnerable Red airbase: I = 1--attack aircraft; I = 2--fighter aircraft performing ABA.
I	CTFMOD	1	-	Several meanings; see definitions of input variables DLT, D2T, TAB10, TAB12, and TAB13.
I	LOCTFF	1	MIMP	Index for movement period of task force.
IAB	ABTACK, CTFMOD, GNAATK, PLBAB, PRTSUM	1	2	Red airbase. IAB = 1 corresponds to the vulnerable Red airbase; IAB = 2, the vulnerable Red airbase.
IATF	ABATCK	1	2	"Is attack on task force planned?" IATF = 1--Red will attack Blue task force later on in clock-time period, IATF = 2--Red will not.
IBAR	MOVRS, MOVTF, PRTSUM	1	NLOC	Barrier IBAR is the barrier between regions IBAR-1 and IBAR.
K	ABATCK	1	3	Criterion for Red aircraft needed to warrant attack. See definition of input variable RARBAB(K).
K	CTFMOD	1	NKRB+1	Type of Red attacker: K = 1 to NKRB correspond to Red bombers; K = NKRB+1--SSMs from Red submarines.
KBA	ABATCK, POWERP, SHPSHP	1	2	Kind of Blue attacker: KBA = 1--attack aircraft, KBA = 2--fighter aircraft performing attack
KBD	PLBAB, PRTSUM	1	NKBDPL	Kind of Blue defender (aircraft) in the Blue land-based air barrier
KBE	ABATCK	1	1	Kind of Blue escort (currently, only fighter aircraft perform the escort mission)
KRA	ABATCK	1	NKRA= NKRB+2	Kind of Red aircraft on vulnerable Red airbase: KRA = 1 to NKRB, bombers; NKRB+1, escort aircraft; NKRB+2, interceptor aircraft
KRB	ABATCK, CTFMOD, GNAATK, PLBAB, PRTSUM	1	NKRB	Kind of Red attacker (bomber).
KRD	ABATCK	1	2	Kind of Red defender of vulnerable Red airbase: KRD = 1--Red escort aircraft on defense; KRD = 2--Red interceptor aircraft
KBS	MOVTF	1	6	Kind of Blue ship: 1--carriers, 2--AAW escort ships, 3--Air-capable ASW escort ships, 4--Non-air-capable ASW escort ships, 5--URG ships, 6--direct support submarines
KBSS	SHPSHP	1	5	Kind of Blue surface ship; as above, for index values 1 through 5
KRS	DDAY, MOVRS, PRTSUM, SHPSHP, SUBSUB	1	NKRS	Kind of Red ship: KRS = 1--torpedo submarines, KRS = 2--missile submarines, KRS = 3,...,NKRS--surface ships.
KRS	POWERP	1	NPPSAM	Kind of Red SAM defending against Blue power projection
KRSS	PRTSUM, SHPSHP	1	NKRSS= NKRS-2	Kind of Red surface ship. Surface ship kinds 1 to NKRSS correspond to ship kinds 3 to NKRS, respectively.
KRSAM	ABATCK	1	NABSAM	Kind of Red SAM defending the vulnerable Red airbase
L	ABATCK, CTFMOD, DDAY, GNAATK, PLBAB, POWERP, SHPSHP, SUBSUB	(=LOCTF)		Region (location) the task force is in; set in MEDMOD to current value of LOCTF
L	PRTSUM	1	NLOC	Region (location)
LB	PLBAB	1	NLOC	Region (location). Used to index Blue land bases corresponding to the regions.
LOC	MOVRS	1	NLOC	Region (location)
LOCTF1	MOVRS	(=LOCTF+1)		Current location of the task force + 1. Used to index input variable PRSM; see its definition.
NKRBP1	DDAY	(=NKRB+1)		See definition of input variable ENACDT.

[a]See Table B-7: Limit Variables, infra, for definitions of the upper limit variables listed here.

Table B-7. Limit Variables

Limit variables indicate the number of kinds available of a resource. Each limit variable should be at least 1.

A. Input Limit Variables

Variable	Program segment(s) in which it appears	Definition	Upper limit with current program dimensioning
MIMP	MOVTF, LOCTFF	Number of movement periods for task force.	6
NABSAM	ABATCK	Number of kinds of Red SAMs defending the vulnerable Red airbase.	2
NKBDPL	PLBAB, PRTSUM	Number of kinds of Blue land based aircraft defending against Red attacking aircraft.	2
NKRB	ABATCK, CTFMOD GNAATK, PLBAB PRTSUM	Number of kinds of Red bombers	3
NKRS[a]	MOVRS, SHPSHP, SUBSUB, PRTSUM	Number of kinds of Red ships (kinds 1 and 2 are torpedo and missile submarines, respectively).	10
NLOC	MOVRS, PRTSUM	Number of possible regions for the task force excluding region zero.	5
NPPSAM	POWERP	Number of kinds of Red SAMs defending against Blue power projection.	2

[a]Variable NKRS should be at least 3.

B. Limit Variables Computed in Program

Variable	Program segment(s) in which it appears	Definition	Value	Comments
NKBS	MOVTF	Number of kinds of Blue ships.	6	There are six kinds of Blue ships: carriers, three kinds of escort ships, URG ships, and direct support submarines.
NKRA	ABATCK	Number of kinds of Red aircraft on the vulnerable Red airbase.	NKRB+2	NKRB kinds of Red bombers, escort aircraft, and interceptor aircraft.
NKRSS	SHPSHP, PRTSUM	Number of kinds of Red surface ships.	NKRS-2	There are NKRS kinds of Red ships: 1 and 2 are submarines, thus kinds 3 to NKRS are surface ships.

Table B-8

Input Variables That Are Used in Several Major Subroutines[a]

Variable	Definition	Major subroutines in which this variable appears
BARLTH(IBAR)	Length of submarine barrier between regions IBAR-1 and IBAR.	MOVRS, MOVTF
BUCAP	The number of sea-based aircraft required to support one CAP station.	ABATCK, CTFMOD, POWERP, SHPSHP
ENACDS(KRS)	Expected number of Blue aircraft destroyed when a shot from a Red ship of type KRS hits a full carrier.	DDAY, SHPSHP
ENACDT(K)	Expected number of Blue aircraft destroyed when an ASM of type K hits a full carrier.	CTFMOD, DDAY
ICTL(IBAR)	Indicator for control of submarine barrier between regions IBAR-1 and IBAR. (See list of input variable definitions for more details.)	MOVRS, MOVTF
PAFCNF	Probability that a (Blue) attack aircraft cannot fly another sortie during a clock time period given that it has already flown during that clock time period.	ABATCK, SHPSHP
PFFCNF	Probability that a (Blue) fighter aircraft cannot fly another sortie in that clock time period given that it has already flown during that clock time period.	ABATCK, CTFMOD, SHPSHP

[a]Excluding resource and limit input variables; see Tables B-4 and B-7, respectively.

TABLE B-9. SELECTED INPUT VARIABLES WITH RESTRICTIONS

PART 1: INPUT VARIABLES THAT MUST BE STRICTLY GREATER THAN ZERO[a,b]

Variable	Indices and Limits[c] (for vector variables)	Program Segment(s) Using This Variable
BARLQ(L)	L=1,NLOC	CTFMOD
D1T(I,KRB)	I=1,2 KRB=1,NKRB	CTFMOD
HRMAAW		CTFMOD
HRMASW		CTFMOD
HRMURG		CTFMOD
HRTAAW		CTFMOD
HRTASW		CTFMOD
HRTURG		CTFMOD
IATKRT(L)[d]	L=1,NLOC	GNAATK
LGTHMP(I)[d]	I=1,MIMP	MOVTF, LOCTFF
MAXTP[d]		MEDMOD
PARK		ABATCK, ATRTAB
THSCAQ(L)	L=1,NLOC	CTFMOD
THSCTQ(L)	L=1,NLOC	CTFMOD
VBT(KRB)	KRB=1,NKRB	CTFMOD
VI		CTFMOD
XNRAB		ABATCK

[a] For vector variables, each component (within the limits stated in the second column) must be strictly greater than zero.

[b] This table does not include limit variables; see Table B-7, *supra*.

[c] The notation L=1,NLOC means that L varies *from* 1 *through* NLOC (inclusive, similarly for other variables. *See* Tables B-6 and B-7, *supra*, for more information.

[d] These integer variables must be at least one (in every appropriate component).

TABLE B-9. SELECTED INPUT VARIABLES WITH RESTRICTIONS

PART 2: INTEGER INPUT VARIABLES THAT CAN TAKE ON VERY FEW VALUES

Variable	Indices and Limits[a] (for vector variables)	Program Segment(s) Using This Variable	Allowable Values for This Variable[b]
IAAED		ABATCK,AIRAIR,ATRTED	0,1
IABAEQ		ABATCK,ATRTAB	1,2,3
IABAW		ABATCK,ATRTSS	1,2
IATRIA		CTFMOD,ATRTIA	1,2
ICTL(IBAR)	IBAR=1,NLOC	MOVRS,MOVTF	0,1,2,3
IDDAC		DDAY	1,2
IDDAS		DDAY	1,2
IKRAS(KRA)	KRA=1,NKRA (NKRA=NKRB+2)	ABATCK	0,1
IPLAED		PLBAB,AIRAIR,ATRTED	0,1
IPPAW		POWERP,ATRTSS	1,2
IRSUBA(L)	L=1,NLOC	CTFMOD	0,1,2
ISSBR		SHPSHP	0,1
ISSRB		SHPSHP	0,1

[a] The notation IBAR=1,NLOC means that IBAR varies <u>from</u> 1 <u>through</u> NLOC (inclusive); similarly for other variables.

[b] For vector variables, each component (within the limits stated in the second column) must take on one of the allowable values, but it is not necessary for all components to have the same value.

TABLE B-9. SELECTED INPUT VARIABLES WITH RESTRICTIONS

PART 3: OTHER RESTRICTIONS

Variable(s)	Relevant Program Segment(s)	Restrictions and Comments
BARLTH(IBAR)	MOVRS, MOVTF	Must be strictly greater than zero if ICTL(IBAR)≠0 for the same value of IBAR (for IBAR=1 through NLOC).
D1T(I,KRB) D2T(I,KRB) VBT(KRB)	CTFMOD	If NKRB≥2, then, for KRB=1 through NKRB-1 and for all I, the relationships D1T(I,KRB)≥D1T(I,KRB+1), D2T(I,KRB)≥D2T(I,KRB+1), and VBT(KRB)≥VBT(KRB+1) must hold (this ensures that different kinds of bombers are ranked in decreasing order of particular types of effectiveness). Also, the method for computing attrition to Red bombers after they launch their ASMs requires that D1T(1,KRB)≥D1T(2,KRB) and D2T(1,KRB)≥D2T(2,KRB) for KRB=1 through NKRB.
FFACA(L) FFACE(L)	ABATCK	For each region (L=1 through NLOC), FFACA(L)+FFACE(L) must not exceed 1. (One cannot allocate more fighter aircraft than are available.)
LTFMP(I)	MOVTF, LOCTFF	See Table B-10: <u>Program Terminations</u>, STOPs 6404 and 6405.
MAXTP	MEDMOD	Must not exceed the dimension limit of (computed) variable LTASKF(ITP); this limit is currently 90. (Variable LTASKF appears in COMMON block COMOUT.)

Table B-10

PROGRAM TERMINATIONS

MEDMOD has several checks built in to stop the program if values are input that make certain program segments meaningless to execute. The corresponding FORTRAN statements are STOP N, where N is the four digit number below. The STOP N statement appears near the end of the program segment indicated.

STOP Number	Program Segment in which it appears	Explanation and Comments
6400	DRIVER	(normal termination)
6401	BARKCK	Zero kinds of penetrators; variable NKRS should be at least 3.
6404	MOVTF	Task force directed to move to a region not adjacent to previous region. No adjacent components of the input array LTFMP should differ by more than unity.
6405	MOVTF	Task force directed to move to a region exceeding the number of regions played. No component of input array LTFMP should be greater than NLOC.
6406	BINOM	Tried to compute binomial probability for $P<0$, $P>1$, or $M>N$.
6407	SHPSHP	Zero kinds of Red surface ships; variable NKRS should be at least 3.
6410	SUBSUB	Zero kinds of Red surface ships; variable NKRS should be at least 3.

TABLE B-11. CORRESPONDENCE BETWEEN MEDMOD INPUTS AND R-245 INPUTS

MEDMOD Input[a]	Corresponding R-245 Input[b]	Order Number Assigned to R-245 Input
AESCAB(IAB)	none	
AEWD	ABWD	27
ASWF	ASWF	11
ATABT(IAB,KRB)	AT	51
BAREAQ(L)	BAREA	9
BARELQ(L)	BAREAL	7
BARLQ(L)	BARL	10
BUCAP	BUCAP	35
CAPMLQ(L)	CAPML	34
CAPMQ(L)	CAPM	33
CAPMR	CAPMR	39
CAPSTQ(L)	CAPSTAR	36
DLIA	DLIA	41
D1T(I,KRB)	D1	54
D2T(I,KRB)	D2	44
ENACDT(K)	none	
ESLR	ESLR	17
ESRQ(L)	ESR	16
FPPL1	FPPL1	57
FPPL2	FPPL2	59
FSTAQ(L)	none	
FSTGAQ(L)	none	

[a] This table lists all of the inputs used in Subroutine CTFMOD as well as the inputs PPSORR and WFPPAS, which are used in Subroutine POWERP. The variables PPSORR and WFPPAS are the only inputs to MEDMOD that both correspond to R-245 inputs and are not used in Subroutine CTFMOD.

[b] All R-245 inputs that correspond to MEDMOD inputs are listed here; there are some R-245 inputs that do not correspond to any MEDMOD input, and so are not listed here.

(continued)

TABLE B-11. (continued)

MEDMOD Input	Corresponding R-245 Input	Order Number Assigned to R-245 Input
HRMAAW	none	
HRMASW	none	
HRMURG	none	
HRTAAW	none	
HRTASW	none	
HRTURG	none	
IATRIA	none	
IRSUBA(L)	none	
NKRB	fixed at 1	
PDIN	PDIN	13
PPFCNF	none	
PKASW	PKASW	12
PKAT1	PKAT1	55
PKDF1	none	
PKIIN	PKIIN	18
PKIN	PKIN	15
PKPLDT(K)	PKPLD	61
PKPL1	PKPL1	58
PKPL2	PKPL2	60
PKSST(K)	PKSS	56
PPSORR(1,L)	SA	62
PPSORR(2,L)	SF	63
PRWLNQ(L)	none	
RS(1,L)	ST	19
RS(2,L)	STG	23
SMALLR	SMALLR	45
SSDAAW	none	
SSDASW	none	
SSDURG	none	
STARQ(L)	STAR	28

(continued)

TABLE B-11. (continued)

MEDMOD Input	Corresponding R-245 Input	Order Number Assigned to R-245 Input
STSALV	STSALV	20
SUBSOR	SUBSOR	22
TAB1ØT(I,K)	TAB1Ø(I)	66
TAB12(I)	TAB12(I)	67
TAB13T(I,K)	TAB13(I)	68
TCAP	TCAP	40
THSCAQ(L)	none	
THSCTQ(L)	none	
TPS	TPS	21
T1	T1	47
T2	T2	48
T3	T3	49
T4	T4	50
UBAEW	UBAEW	25
UBAEWL	UBAEWL	26
UBASW	UBASW	8
UBASWL	UBASWL	6
VBT(K)	VB	53
VCAP	VCAP	38
VI	VI	46
WFMAAW	none	
WFMASW	none	
WFMPLT	none	
WFMURG	none	
WFTAAW	none	
WFTASW	none	
WFTPLT	none	
WFTURG	none	
WFPPAS(1,L)	WA	64
WFPPAS(2,L)	WF	65

(continued)

TABLE B-11. (concluded)

MEDMOD Input	Corresponding R-245 Input	Order Number Assigned to R-245 Input
WRLNDQ(L)	none	
WVSIZ	WVSIZ	42
XAEW	XAEW	Blue Resource
XAEWLQ(L)	XAEWL	Blue Resource
XASW	XASW	Blue Resource
XASWLQ(L)	XASWL	Blue Resource
XATTCK	none[c]	
XEAAW	XEAAW	Blue Resource
XEASWA	XEASWA	Blue Resource
XEASWN	XEASWN	Blue Resource
XFGHTR	none[c]	
XPLAT	XPLAT	Blue Resource
XURGS	none	
ZLAMPF	ZLAMPF	14
ZMPATT(K)	ZMPAT	52
ZMPCAP	ZMPCAP	37
ZMPDLI	ZMPDLI	43
ZMPESC	none	
ZMPSTG	ZMPSTG	24

[c]There is a rough correspondence between the MEDMOD inputs XATTCK and XFGHTR, and the R-245 inputs SPA (R-245 order number 4), SPF (R-245 order number 3), and SPLAT (R-245 order number 5).

TABLE B-12. INDEX OF MAJOR VARIABLES USED IN MEDMOD

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
AAAEDA(KRD)	INPUT (PARAMETER)	ABATCK
AAAEDF(KRD)	INPUT (PARAMETER)	ABATCK
AAAEED(KBE)	INPUT (PARAMETER)	ABATCK
AACA	INPUT (PARAMETER)	ABATCK
AAPAJO(KBA)	INPUT (PARAMETER)	ABATCK
AAPDDA(KRD)	INPUT (PARAMETER)	ABATCK
AAPDDE(KRD)	INPUT (PARAMETER)	ABATCK
AAPDED(KBE)	INPUT (PARAMETER)	ABATCK
AAPKAD(KBA,KRD)	INPUT (PARAMETER)	ABATCK
AAPKDA(KRD,KBA)	INPUT (PARAMETER)	ABATCK
AAPKDE(KRD,KBE)	INPUT (PARAMETER)	ABATCK
AAPKED(KBE,KRD)	INPUT (PARAMETER)	ABATCK
AASRAA(L)	INPUT (PARAMETER)	ABATCK
AASRED	INPUT (PARAMETER)	ABATCK
AASRFA(L)	INPUT (PARAMETER)	ABATCK
AASRFE(L)	INPUT (PARAMETER)	ABATCK
AASRID	INPUT (PARAMETER)	ABATCK
ABANM(KRSAM)	INPUT (RESOURCE)	ABATCK
ABAVLS(KRSAM)	INPUT (PARAMETER)	ABATCK
ABCAS	INPUT (PARAMETER)	ABATCK
ABESGS(KBA)	INPUT (PARAMETER)	ABATCK
ABFASS(KBA)	INPUT (PARAMETER)	ABATCK
ABFSM(KBA)	INPUT (PARAMETER)	ABATCK
ABFVS(KRSAM)	INPUT (PARAMETER)	ABATCK
ABPDA(KBA)	INPUT (PARAMETER)	ABATCK
ABPDS(KRSAM)	INPUT (PARAMETER)	ABATCK
ABPKA(KRSAM)	INPUT (PARAMETER)	ABATCK
ABPKS(KRSAM,KBA)	INPUT (PARAMETER)	ABATCK
ABPSA(KBA,KRSAM)	INPUT (PARAMETER)	ABATCK
ABRSAM(KRSAM)	INPUT (RESOURCE)	ABATCK
ABTSC(KRSAM)	INPUT (PARAMETER)	ABATCK
ABVGSS(KRSAM)	INPUT (PARAMETER)	ABATCK
AESCAB(IAB)	INPUT (RESOURCE)	ABATCK CTFMOD GNAATK PLBAB PRTSUM

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
AFWD	INPUT (PARAMETER)	CTFMOD
AINTCT	INPUT (RESOURCE)	ABATCK PRTSUM
ASWF	INPUT (PARAMETER)	CTFMOD
ATABT(IAB,KRB)	INPUT (RESOURCE)	ABATCK CTFMOD GNAATK PLBAB PRTSUM
ATSORU	LABELED COMMON: COMSOR	MEDMOD ABATCK CTFMOD POWERP SHPSHP
ATTCKI	LABELED COMMON: COMCTF	MEDMOD ABATCK CTFMOD DDAY POWERP SHPSHP
ATTWGT	INPUT (PARAMETER)	MOVTF
AVAILF(L,IAB)	INPUT (PARAMETER)	GNAATK
AVAILT(L,IAB,KRB)	INPUT (PARAMETER)	GNAATK
AVALED(L,IATF)	INPUT (PARAMETER)	ABATCK
AWRCBB	INPUT (PARAMETER)	MOVRS
BACCDW(KBS)	INPUT (PARAMETER)	MOVTF
BACPCK(KBS)	INPUT (PARAMETER)	MOVTF
BAREAQ(L)	INPUT (PARAMETER)	CTFMOD
BARELQ(L)	INPUT (PARAMETER)	CTFMOD
BARLQ(L)	INPUT (PARAMETER)	CTFMOD
BARLTH(IBAR)	INPUT (PARAMETER)	MOVRS MOVTF
BECDW(KBS)	INPUT (PARAMETER)	MOVTF
BEDW(KRS)	INPUT (PARAMETER)	MOVRS
BMR(IAB,KRB)	LABELED COMMON: COMGA	CTFMOD GNAATK PLBAB
BMTMIN(L)	INPUT (PARAMETER)	GNAATK
BSIBAR(IBAR)	INPUT (RESOURCE)	MOVRS PRTSUM
BSSNDS	INPUT (RESOURCE)	MOVTF PRTSUM SUBSUB
BUCAP	INPUT (PARAMETER)	ABATCK CTFMOD POWERP SHPSHP
CACDWO	INPUT (PARAMETER)	MOVTF
CAPMLQ(L)	INPUT (PARAMETER)	CTFMOD
CAPMQ(L)	INPUT (PARAMETER)	CTFMOD
CAPMR	INPUT (PARAMETER)	CTFMOD
CAPSTQ(L)	INPUT (PARAMETER)	CTFMOD
CPAGV	INPUT (PARAMETER)	MOVTF
CPBPK(KBS)	INPUT (PARAMETER)	MOVTF
CPBSCK(KRS)	INPUT (PARAMETER)	MOVRS

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
CPRPK(KRS)	INPUT (PARAMETER)	MOVRS
CPRSCK(KBS)	INPUT (PARAMETER)	MOVTF
CSCDWO	INPUT (PARAMETER)	MOVTF
CWPPAS	LABELED COMMON: COMOUT	MEDMOD POWERP PRTSUM
CWTPTF	LABELED COMMON: COMOUT	MEDMOD PRTSUM
DDFAC(KRS)	INPUT (PARAMETER)	DDAY
DDPKC(KRS)	INPUT (PARAMETER)	DDAY
DDPKS(KRS)	INPUT (PARAMETER)	DDAY
DDRKAA(KRS)	INPUT (PARAMETER)	DDAY
DDRKBA(KRS)	INPUT (PARAMETER)	DDAY
DDRSA(KRS)	INPUT (PARAMETER)	DDAY
DDSPA(KRS)	INPUT (PARAMETER)	DDAY
DLIA	INPUT (PARAMETER)	CTFMOD
D1T(I,KRB)	INPUT (PARAMETER)	CTFMOD
D2T(I,KRB)	INPUT (PARAMETER)	CTFMOD
ENACDS(KRS)	INPUT (PARAMETER)	DDAY SHPSHP
ENACDT(K)	INPUT (PARAMETER)	CTFMOD
ENACDT(NKRBP1)	INPUT (PARAMETER)	DDAY
ESC(IAB)	LABELED COMMON: COMGA	CTFMOD GNAATK PLBAB
ESLR	INPUT (PARAMETER)	CTFMOD
ESRQ(L)	INPUT (PARAMETER)	CTFMOD
FAACA(L)	INPUT (PARAMETER)	ABATCK
FACDB(KRA,IATF)	INPUT (PARAMETER)	ABATCK
FFACA(L)	INPUT (PARAMETER)	ABATCK
FFACE(L)	INPUT (PARAMETER)	ABATCK
FGHTRI	LABELED COMMON: COMCTF	MEDMOD ABATCK CTFMOD DDAY POWERP SHPSHP
FHSK(I)	INPUT (PARAMETER)	ABATCK
FM3(KBS)	INPUT (PARAMETER)	MOVTF
FPPL1	INPUT (PARAMETER)	CTFMOD
FPPL2	INPUT (PARAMETER)	CTFMOD
FSTAQ(L)	INPUT (PARAMETER)	CTFMOD
FSTGAC(L)	INPUT (PARAMETER)	CTFMOD

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
FTSORU	LABELED COMMON: COMSOR	MEDMOD ABATCK CTFMOD POWERP SHPSHP
HRMAAW	INPUT (PARAMETER)	CTFMOD
HRMASW	INPUT (PARAMETER)	CTFMOD
HRMURG	INPUT (PARAMETER)	CTFMOD
HRTAAW	INPUT (PARAMETER)	CTFMOD
HRTASW	INPUT (PARAMETER)	CTFMOD
HRTURG	INPUT (PARAMETER)	CTFMOD
I	INDEXING	ABATCK CTFMOD GNAATK MOVTF
IAADA	INPUT (PARAMETER)	ABATCK
IAAED	INPUT (PARAMETER)	ABATCK
IAB	INDEXING	ABATCK CTFMOD GNAATK PLBAB PRTSUM
IABAEQ	INPUT (PARAMETER)	ABATCK
IABAF	INPUT (PARAMETER)	ABATCK
IABAW	INPUT (PARAMETER)	ABATCK
IATF	INDEXING	ABATCK
IATKRT(L)	INPUT (PARAMETER)	GNAATK
IATRIA	INPUT (PARAMETER)	CTFMOD
IBAR	INDEXING	MOVRS MOVTF PRTSUM
ICTL(IBAR)	INPUT (PARAMETER)	MOVRS MOVTF
IDDAC	INPUT (PARAMETER)	DDAY
IDDAS	INPUT (PARAMETER)	DDAY
IGO	LABELED COMMON: COMIGO	MEDMOD TIMET INP
IKRAS(KRA)	INPUT (PARAMETER)	ABATCK
IPLADA	INPUT (PARAMETER)	PLBAB
IPLAED	INPUT (PARAMETER)	PLBAB
IPPAF	INPUT (PARAMETER)	POWERP
IPPAW	INPUT (PARAMETER)	POWERP
IPSUBA(L)	INPUT (PARAMETER)	CTFMOD
ISSBR	INPUT (PARAMETER)	SHPSHP
ISSRB	INPUT (PARAMETER)	SHPSHP
ITP	(TIME PERIOD)	DRIVER MEDMOD ADDMOE GNAATK LOCTFF MOVRS MOVTF POWERP PRTSUM SHPSHP
K	INDEXING	ABATCK CTFMOD GNAATK PLBAB

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
KBA	INDEXING	ABATCK POWERP SHPSHP
KBD	INDEXING	PLBAB PRTSUM
KBE	INDEXING	ABATCK
KBS	INDEXING	MOVTF
KRA	INDEXING	ABATCK PLBAB
KRB	INDEXING	ABATCK CTFMOD GNAATK PLBAB PRTSUM
KRD	INDEXING	ABATCK
KRS	INDEXING	DDAY MOVRS POWERP PRTSUM SHPSHP SUBSUB
KRSAM	INDEXING	ABATCK
KRSS	INDEXING	PRTSUM SHPSHP
L	INDEXING	ABATCK CTFMOD DDAY GNAATK MOVRS PLBAB POWERP PRTSUM SHPSHP SUBSUB
LB	INDEXING	PLBAB PRTSUM
LC	INDEXING	PRTSUM
LGTHMP(I)	INPUT (PARAMETER)	LOCTFF MOVTF
LOC	INDEXING	MOVRS
LOCTF	INDEXING	MEDMOD MOVRS MOVTF
LOCTF1	INDEXING	MOVRS
LTASKF(ITP)	LABELED COMMON: COMOUT	DRIVER MEDMOD
LTFMP(I)	INPUT (PARAMETER)	LOCTFF MOVTF
MAXTP	INPUT (LIMIT)	DRIVER MEDMOD
MIMP	INPUT (LIMIT)	MEDMOD LOCTFF MOVTF
NABSAM	INPUT (LIMIT)	ABATCK
NEPD	INPUT (PARAMETER)	TIMET INP
NKBDPL	INPUT (LIMIT)	PLBAB PRTSUM
NKBS	COMPUTED LIMIT	MOVTF
NKRA	COMPUTED LIMIT	ABATCK
NKRB	INPUT (LIMIT)	ABATCK CTFMOD GNAATK PLBAB PRTSUM
NKRBP1	INDEXING	DDAY
NKRS	INPUT (LIMIT)	MOVRS PRTSUM SHPSHP SUBSUB
NKRSS	COMPUTED LIMIT	PRTSUM SHPSHP
NLOC	INPUT (LIMIT)	MOVRS PRTSUM

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
NPPSAM	INPUT (LIMIT)	POWERP
NTPSIM	LABELED COMMON: COMOUT	MEDMOD ADDMOE
NTPSLA	LABELED COMMON: COMGA	MEDMOD ABATCK GNAATK
PAFCNF	INPUT (PARAMETER)	ABATCK SHPSHP
PARK	INPUT (PARAMETER)	ABATCK
PASS(I)	INPUT (PARAMETER)	ABATCK
PBDRN(I)	INPUT (PARAMETER)	ABATCK
PBDRS(I)	INPUT (PARAMETER)	ABATCK
PBKRN(I)	INPUT (PARAMETER)	ABATCK
PBKRS(I)	INPUT (PARAMETER)	ABATCK
PDIN	INPUT (PARAMETER)	CTFMOD
PFFCNF	INPUT (PARAMETER)	ABATCK CTFMOD SHPSHP
PKASW	INPUT (PARAMETER)	CTFMOD
PKAT1	INPUT (PARAMETER)	CTFMOD
PKDF1	INPUT (PARAMETER)	CTFMOD
PKIIN	INPUT (PARAMETER)	CTFMOD
PKIN	INPUT (PARAMETER)	CTFMOD
PKPLDT(K)	INPUT (PARAMETER)	CTFMOD
PKPL1	INPUT (PARAMETER)	CTFMOD
PKPL2	INPUT (PARAMETER)	CTFMOD
PKSST(K)	INPUT (PARAMETER)	CTFMOD
PLAEDA(KBD)	INPUT (PARAMETER)	PLBAB
PLAEDE(KBD)	INPUT (PARAMETER)	PLBAB
PLAEED	INPUT (PARAMETER)	PLBAB
PLBLBD(KBD,LB)	INPUT (RESOURCE)	PLBAB PRTSUM
PLCA(L)	INPUT (PARAMETER)	PLBAB
PLFDLL(LB,L,KBD)	INPUT (PARAMETER)	PLBAB
PLPAJO(KRA)	INPUT (PARAMETER)	PLBAB
PLPDDA(KBD)	INPUT (PARAMETER)	PLBAB
PLPDDE(KBD)	INPUT (PARAMETER)	PLBAB
PLPDED	INPUT (PARAMETER)	PLBAB
PLPKAD(KRA,KBD)	INPUT (PARAMETER)	PLBAB
PLPKDA(KBD,KRA)	INPUT (PARAMETER)	PLBAB

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
PLPKDE(KBD)	INPUT (PARAMETER)	PLBAB
PLPKED(KBD)	INPUT (PARAMETER)	PLBAB
PPAEGS(KBA)	INPUT (PARAMETER)	POWERP
PPANMS(KRSAM)	INPUT (RESOURCE)	POWERP
PPAVLS(KRS,L)	INPUT (PARAMETER)	POWERP
PPAVSS(KRS)	INPUT (PARAMETER)	POWERP
PPCAL(L)	INPUT (PARAMETER)	POWERP
PPFASM(KBA)	INPUT (PARAMETER)	POWERP
PPFASS(KBA)	INPUT (PARAMETER)	POWERP
PPFSVS(KRS)	INPUT (PARAMETER)	POWERP
PPPDAS(KBA)	INPUT (PARAMETER)	POWERP
PPPDSA(KRS)	INPUT (PARAMETER)	POWERP
PPPKAS(KRS)	INPUT (PARAMETER)	POWERP
PPPKSA(KRS,KBA)	INPUT (PARAMETER)	POWERP
PPPSAS(KBA,KRS)	INPUT (PARAMETER)	POWERP
PPRSAM(KRSAM)	INPUT (RESOURCE)	POWERP
PPSORR(KBA,L)	INPUT (PARAMETER)	POWERP
PPSORT	LABELED COMMON: COMOUT	MEDMOD POWERP PRTSUM
PPTSCS(KRS)	INPUT (PARAMETER)	POWERP
PRSM(KRS,LOC,LOCTF1)	INPUT (PARAMETER)	MOVRS
PRWLNQ(L)	INPUT (PARAMETER)	CTFMOD
RACCDW(KRS)	INPUT (PARAMETER)	MOVRS
RACPCK(KRS)	INPUT (PARAMETER)	MOVRS
RARBAB(K)	INPUT (PARAMETER)	ABATCK
RECDW(KRS)	INPUT (PARAMETER)	MOVRS
RFDW(KBS)	INPUT (PARAMETER)	MOVTF
RS(1,L)	INPUT (RESOURCE)	CTFMOD DDAY MOVRS PRTSUM SUBSUB
RS(2,L)	INPUT (RESOURCE)	CTFMOD DDAY MOVRS PRTSUM SUBSUB
RS(KRS,L),KRS≥3	INPUT (RESOURCE)	DDAY MOVRS PRTSUM SHPSHP SUBSUB
RSIBAR(IBAR)	INPUT (RESOURCE)	MOVTF PRTSUM
SBFBCF	INPUT (PARAMETER)	SUBSUB
SBFBCS	INPUT (PARAMETER)	SUBSUB
SBFRFA(L)	INPUT (PARAMETER)	SUBSUB

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
SBFRFC	INPUT (PARAMETER)	SUBSUB
SBFRSA(L)	INPUT (PARAMETER)	SUBSUB
SBFRSC	INPUT (PARAMETER)	SUBSUB
SBPBDF	INPUT (PARAMETER)	SUBSUB
SBPBDS	INPUT (PARAMETER)	SUBSUB
SBPBKF	INPUT (PARAMETER)	SUBSUB
SBPBKS	INPUT (PARAMETER)	SUBSUB
SBPFDB	INPUT (PARAMETER)	SUBSUB
SBPFKB	INPUT (PARAMETER)	SUBSUB
SBPSDB	INPUT (PARAMETER)	SUBSUB
SBPSKB	INPUT (PARAMETER)	SUBSUB
SCK31	LABELED COMMON: BARSCK	MOVTF
SCK32	LABELED COMMON: BARSCK	MOVTF
SHEL	INPUT (RESOURCE)	ABATCK
SMALLR	INPUT (PARAMETER)	CTFMOD
SSBACR(KRSS)	INPUT (PARAMETER)	SHPSHP
SSCFA	INPUT (PARAMETER)	SHPSHP
SSDAAW	INPUT (PARAMETER)	CTFMOD
SSDASW	INPUT (PARAMETER)	CTFMOD
SSDURG	INPUT (PARAMETER)	CTFMOD
SSFBAK(KBA,KRSS)	INPUT (PARAMETER)	SHPSHP
SSFRSV(KRSS,L)	INPUT (PARAMETER)	SHPSHP
SSPBDR	INPUT (PARAMETER)	SHPSHP
SSPBKR	INPUT (PARAMETER)	SHPSHP
SSPRDB	INPUT (PARAMETER)	SHPSHP
SSPRKB	INPUT (PARAMETER)	SHPSHP
SSPRKC	INPUT (PARAMETER)	SHPSHP
STARQ(L)	INPUT (PARAMETER)	CTFMOD
STSALV	INPUT (PARAMETER)	CTFMOD
SUBSOR	INPUT (PARAMETER)	CTFMOD
TAB10T(I,K)	INPUT (PARAMETER)	CTFMOD
TAB12(I)	INPUT (PARAMETER)	CTFMOD
TAB13T(I,K)	INPUT (PARAMETER)	CTFMOD

continued

TABLE B-12. (continued)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
TCAP	INPUT (PARAMETER)	CTFMOD
THSCAQ(L)	INPUT (PARAMETER)	CTFMOD
THSCTG(L)	INPUT (PARAMETER)	CTFMOD
TPAS	INPUT (PARAMETER)	MOVTF
TPS	INPUT (PARAMETER)	CTFMOD
T1	INPUT (PARAMETER)	CTFMOD
T2	INPUT (PARAMETER)	CTFMOD
T3	INPUT (PARAMETER)	CTFMOD
T4	INPUT (PARAMETER)	CTFMOD
UBAEWL	INPUT (PARAMETER)	CTFMOD
UBAFW	INPUT (PARAMETER)	CTFMOD
UBASWL	INPUT (PARAMETER)	CTFMOD
UBASW	INPUT (PARAMETER)	CTFMOD
VBT(K)	INPUT (PARAMETER)	CTFMOD
VCAP	INPUT (PARAMETER)	CTFMOD
VI	INPUT (PARAMETER)	CTFMOD
WFMAAW	INPUT (PARAMETER)	CTFMOD
WFMASW	INPUT (PARAMETER)	CTFMOD
WFMPLT	INPUT (PARAMETER)	CTFMOD
WFMURG	INPUT (PARAMETER)	CTFMOD
WFPPAS(KBA,L)	INPUT (PARAMETER)	POWERP
WFTAAW	INPUT (PARAMETER)	CTFMOD
WFTASW	INPUT (PARAMETER)	CTFMOD
WFTFL(L)	INPUT (PARAMETER)	PRTSUM
WFTPLT	INPUT (PARAMETER)	CTFMOD
WFTURG	INPUT (PARAMETER)	CTFMOD
WRLNDO(L)	INPUT (PARAMETER)	CTFMOD
WTFCBO	INPUT (PARAMETER)	MOVTF
WVSIZ	INPUT (PARAMETER)	CTFMOD
XAFW	INPUT (RESOURCE)	CTFMOD
XAEWLQ(L)	INPUT (RESOURCE)	CTFMOD
XASW	INPUT (RESOURCE)	CTFMOD
XASWLQ(L)	INPUT (RESOURCE)	CTFMOD

continued

TABLE B-12. (concluded)

VARIABLE NAME	TYPE OF VARIABLE	PROGRAM SEGMENT(S) USING THIS VARIABLE
XATTCK	INPUT (RESOURCE)	ABATCK CTFMOD DDAY POWERP PRTSUM SHPSHP
XCAPST	LABELED COMMON: COMCTF	MEDMOD ABATCK CTFMOD POWERP SHPSHP
XEAAW	INPUT (RESOURCE)	ADDMOE CTFMOD DDAY MOVTF PRTSUM SHPSHP
XEASWA	INPUT (RESOURCE)	ADDMOE CTFMOD DDAY MOVTF PRTSUM SHPSHP
XEASWN	INPUT (RESOURCE)	ADDMOE CTFMOD DDAY MOVTF PRTSUM SHPSHP
XEFFCM	LABELED COMMON: COMCTF	MEDMOD ABATCK ADDMOE CTFMOD DDAY MOVTF POWERP PRTSUM SHPSHP
XFGHTR	INPUT (RESOURCE)	ABATCK CTFMOD DDAY POWERP PRTSUM SHPSHP
XIA(L)	INPUT (PARAMETER)	ABATCK
XIE(L)	INPUT (PARAMETER)	ABATCK
XNRAB	INPUT (PARAMETER)	ABATCK
XPLAT	INPUT (RESOURCE)	DRIVER ADDMOE CTFMOD DDAY MOVTF PRTSUM SHPSHP
XURGS	INPUT (RESOURCE)	ADDMOE CTFMOD DDAY MOVTF PRTSUM SHPSHP
ZLAMPF	INPUT (PARAMETER)	CTFMOD
ZMPATT(K)	INPUT (PARAMETER)	CTFMOD
ZMPCAP	INPUT (PARAMETER)	CTFMOD
ZMPDLI	INPUT (PARAMETER)	CTFMOD
ZMPESC	INPUT (PARAMETER)	CTFMOD
ZMPSTG	INPUT (PARAMETER)	CTFMOD

APPENDIX C

DEFINITIONS OF INPUTS AND SAMPLE OUTPUT OF INP

DEFINITIONS OF INPUTS AND SAMPLE OUTPUT OF INP

The following is the output produced by Overlay INP given an entirely hypothetical, unclassified data base. This output serves three purposes. First, and most importantly, it gives definitions for all inputs to MEDMOD in alphabetical order.[1] Second, it shows a potential user of MEDMOD what the output of INP (i.e., what the output of the inputs) looks like.[2] Third, it provides a hypothetical data base for testing purposes. Appendix D gives the remainder of the outputs of MEDMOD (i.e., the output of the results) based on this hypothetical data base.

---

[1]Note that the alphabetizing rule used by the CDC-6400 computer ranks blanks after letters, not before them. Thus, for example, RS follows RSIBAR and UBAEW follows UBAEWL on this output.

[2]Note that INP displays all TIMET input changes first, before it displays the initial input values. In the sample output of INP displayed here, some of the entries of two input arrays are to be changed during the run—the entries of AVAILE(L,1) for L = 1, ..., 5 are to be changed from their initial values to 0.1, 0.2, 0.4, 0.8, and 0.8, respectively, at the start of time period 3, and the entries of ATABT(I,1) for I = 1,2 are to be incremented by 5.0 and 5.0, respectively, at the start of time period 8.

```
TIME-T=    3        VARIABLE AVAILE 0    0  1  0---VALUES ARE BELOW
          .1000       .2000       .4000       .8000       .8000       0.

TIME-T=    8INCREM VARIABLE ATABT  0    0  1  0---VALUES ARE BELOW
          5.000       5.000      0.          0.          0.          0.

VARIABLE ---- AAAEDA(  2,   0,   0)
                                            (KRD) AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION TO 1.0)
                                            THAT A RED DEFENDER OF KIND KRD CAN POTENTIALLY MAKE AGAINST BLUE ABA
                                            AIRCRAFT.
                                            **NOTE**INPUT VARIABLES WHOSE NAMES START WITH AA ARE USED IN SUBROU-
                                              TINE ABATCK, ESPECIALLY THE AIR-TO-AIR PORTIONS.
         0.          3.000

VARIABLE ---- AAAEDE(  2,   0,   0)
                                            (KRD) AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS THAT A RED DEFENDER OF
                                            KIND KRD CAN POTENTIALLY MAKE AGAINST BLUE ABA ESCORT AIRCRAFT.
         0.          1.000

VARIABLE ---- AAAEED(  1,   0,   0)
                                            (KBE) AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS THAT A BLUE ABA ESCORT
                                            AIRCRAFT OF KIND KBE CAN SHOOT AT A RED DEFENDER (AIRCRAFT).
          5.000

VARIABLE ---- AACA  (  1,   0,   0)
                                            NUMBER OF AIR-TO-AIR COMBAT AREAS USED IN DEFENSE OF THE VULNERABLE
                                            RED AIRBASE.
          6.000

VARIABLE ---- AAPAJO(  2,   0,   0)
                                            (KBA) PROPORTION OF BLUE ABA AIRCRAFT OF KIND KBA THAT, WHEN ENGAGED
                                            BY A RED DEFENDER, JETTISON THEIR ORDNANCE AND RETURN FIRE.
          .1000       .1200

VARIABLE ---- AAPDDA(  2,   0,   0)
                                            (KRD) PROBABILITY THAT A RED DEFENDER OF KIND KRD WILL DETECT A BLUE
                                            ATTACKER.
          .6400       .7400

VARIABLE ---- AAPDDE(  2,   0,   0)
                                            (KRD) PROBABILITY THAT A RED DEFENDER OF KIND KRD WILL DETECT A BLUE
                                            ABA ESCORT AIRCRAFT.
          .5200       .6200

VARIABLE ---- AAPDED(  1,   0,   0)
                                            (KBE) PROBABILITY THAT A BLUE ABA ESCORT AIRCRAFT OF KIND KBE WILL
                                            DETECT A RED DEFENDER.
          .4200
```

VARIABLE ---- AAPKAD(2, 2, 0)

(KBA,KRD) PROBABILITY THAT AN ATTACKER OF KIND KBA WILL KILL A DEFENDER OF KIND KRD, IF ENGAGED.

.2100 .2300
.3100 .3300

VARIABLE ---- AAPKDA(2, 2, 0)

(KRD,KBA) PROBABILITY THAT A DEFENDER OF KIND KRD WILL KILL AN ATTACKER OF KIND KBA, IF ENGAGED.

.5000E-01 .6000E-01
.3000E-01 .4000E-01

VARIABLE ---- AAPKDE(2, 1, 0)

(KRD,KBE) PROBABILITY THAT A DEFENDER OF KIND KRD WILL KILL AN ESCORT OF KIND KBE, IF ENGAGED.

.1000E-01
.2000E-01

VARIABLE ---- AAPKED(1, 2, 0)

(KBE,KRD) PROBABILITY THAT AN ESCORT OF KIND KBE WILL KILL A DEFENDER OF KIND KRD, IF ENGAGED.

.2700 .1700

VARIABLE ---- AASRAA(5, 0, 0)

(L) SORTIE RATE OF BLUE ATTACK AIRCRAFT (FROM CARRIER) ON AIRBASE ATTACK WHEN THE TASK FORCE IS IN LOCATION L. (SORTIES FLOWN PER TIME PERIOD)

.5600 .6600 .7600 .8600 .8600

VARIABLE ---- AASRED(1, 0, 0)

SORTIE RATE OF RED ESCORT AIRCRAFT ON AIRBASE DEFENSE MISSION.

1.150

VARIABLE ---- AASRFA(5, 0, 0)

(L) SORTIE RATE FOR BLUE FIGHTER AIRCRAFT DOING AIRBASE ATTACK WHEN THE TASK FORCE IS IN LOCATION L.

.1700 .5700 .6700 .7700 .8700

VARIABLE ---- AASRFE(5, 0, 0)

(L) SORTIE RATE FOR BLUE FIGHTER AIRCRAFT DOING AIRBASE ATTACK ESCORT WHEN THE TASK FORCE IS IN LOCATION L.

.7800 1.000 1.000 1.000 1.000

VARIABLE ---- AASRID(1, 0, 0)

SORTIE RATE OF RED INTERCEPTOR AIRCRAFT ON AIRBASE DEFENSE MISSION.

1.000

```
VARIABLE ---- ABANM (  2,   0,   0)
                                        (KRSAM) ACTUAL NUMBER OF MISSILES FOR RED SAMS OF KIND KRSAM DEFENDING
                                        THE VULNERABLE RED AIRBASE.
                                        **NOTE**INPUT VARIABLES WHOSE NAMES START WITH AB ARE USED IN THE SAM
                                          PORTION OF SUBROUTINE ABATCK.
         1000.        0.

VARIABLE ---- ABAVLS(  2,   0,   0)
                                        (KRSAM) AVAILABILITY FACTOR FOR SAMS DEFENDING THE VULNERABLE RED AIR-
                                        BASE.  KRSAM=1,...,NABSAM
         .5100        .7100

VARIABLE ---- ABCAS (  1,   0,   0)
                                        NUMBER OF COMBAT AREAS FOR THE BLUE ABA AIRCRAFT/RED SAM INTERACTION.
                                        THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         7.000

VARIABLE ---- ABESGS(  2,   0,   0)
                                        (KBA) AVERAGE NUMBER OF ADDITIONAL GROUND TARGETS THAT AN AIRCRAFT OF
                                        KIND KBA CAN POTENTIALLY MAKE AFTER HAVING ENGAGED A SAM.
                                        THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         1.800        1.900

VARIABLE ---- ABFASS(  2,   0,   0)
                                        (KBA) FRACTION OF BLUE ABA AIRCRAFT OF KIND KBA THAT CAN DO SAM SUP-
                                        PRESSION.
                                        THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .5900        .7900

VARIABLE ---- ABFSM (  2,   0,   0)
                                        (KBA) FRACTION OF BLUE ABA AIRCRAFT OF KIND KBA THAT ARE INVULNERABLE
                                        TO SAMS (PERHAPS DUE TO STANDOFF MUNITIONS).
                                        THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .9000E-01    .1900

VARIABLE ---- ABFVS (  2,   0,   0)
                                        (KRSAM) FRACTION OF SAMS VULNERABLE TO SAM SUPPRESSORS. (RED SAMS DEF-
                                        ENDING THE VULNERABLE RED AIRBASE.)
                                          KRSAM=1,...,NABSAM
       0.             .5300

VARIABLE ---- ABPDA (  2,   0,   0)
                                        (KBA) PROBABILITY THAT A SAM SUPPRESSOR WILL DETECT A SAM.
                                          KBA=1 FOR BLUE ATTACK AIRCRAFT,  KBA=2 FOR BLUE FIGHTERS.
                                        THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .8100        .8200
```

```
VARIABLE ---- ABPDS (  2,   0,   0)
                                       (KRSAM) PROBABILITY THAT A SAM WILL DETECT AN AIRCRAFT.
                                         KRSAM=1,...,NABSAM
                                       THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .7800        .8800

VARIABLE ---- ABPKA (  2,   0,   0)
                                       (KRSAM) PROBABILITY THAT A SAM SUPPRESSOR WILL KILL A SAM.
                                         KRSAM=1,...,NABSAM
                                       THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .1100        .1200

VARIABLE ---- ABPKS (  2,   2,   0)
                                       (KRSAM,KBA) PROBABILITY THAT A SAM WILL KILL AN AIRCRAFT.
                                         KRSAM=1,...,NABSAM
                                         KBA=1 FOR BLUE ATTACK AIRCRAFT,  KBA=2 FOR BLUE FIGHTERS.
                                       THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .1300E-01    .2300E-01
         .4300E-01    .5300E-01

VARIABLE ---- ABPSA (  2,   2,   0)
                                       (KBA,KRSAM) PROBABILITY THAT A SAM SUPPRESSOR WILL SUPPRESS A SAM.
                                         KRSAM=1,...,NABSAM
                                         KBA=1 FOR BLUE ATTACK AIRCRAFT,  KBA=2 FOR BLUE FIGHTERS.
                                       THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .2100        .3100
         .2500        .3500

VARIABLE ---- ABRSAM(  2,   0,   0)
                                       (KRSAM) INITIAL NUMBER OF RED SAMS OF KIND KRSAM DEFENDING THE VULNER-
                                       ABLE RED AIRBASE (KRSAM=1,...,NABSAM).
         20.00        10.00

VARIABLE ---- ABTSC (  2,   0,   0)
                                       (KRSAM) TOTAL SHOTS PER CYCLE FOR A SAM.
                                         KRSAM=1,...,NABSAM
                                       THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         2.000        6.000

VARIABLE ---- ABVGSS(  2,   0,   0)
                                       (KRSAM) AVERAGE NUMBER OF SHOTS PER FIRE CONTROL CENTER OF KIND KRSAM
                                       PER ATTACK AIRCRAFT.
                                       THIS INPUT IS USED WHEN BLUE ATTACKS THE VULNERABLE RED AIRBASE.
         .5000        1.000

VARIABLE ---- AESCAB(  2,   0,   0)
                                       (I) NUMBER OF RED ESCORTS STATIONED ON NOTIONAL RED AIRBASE I --
                                       I=1 IS THE VULNERABLE AIRBASE, I=2 IS THE INVULNERABLE AIRBASE.
         100.0        100.0
```

```
VARIABLE ---- AEWD  (  1,   0,   0)
                                        THE RADAR DETECTION RANGE FOR AEW AIRCRAFT. NO DISTINCTION IS MADE
                                        BETWEEN SEA- AND LAND-BASED ASSETS. (NMI)
         220.0

VARIABLE ---- AINTCT(  1,   0,   0)
                                        INITIAL NUMBER OF RED INTERCEPTOR AIRCRAFT (ON THE VULNERABLE AIRBASE)
         50.00

VARIABLE ---- ASWF  (  1,   0,   0)
                                        A PARAMETER USED TO MODEL THE INCREASING PROBABILITY OF DETECTING A
                                        SUBMARINE TRAVERSING THE OUTER ASW SCREEN AS THE SCREEN WIDTH
                                        INCREASES.
         .5000E-01

VARIABLE ---- ATABT (  2,   3,   0)
                                        (I,K) THE NUMBER OF RED BOMBERS OF TYPE K STATIONED ON RED AIRBASE I,
                                        I=1 IS THE VULNERABLE AIRBASE, I=2 IS THE INVULNERABLE AIRBASE.
         40.00        20.00        20.00
         40.00        20.00        20.00

VARIABLE ---- ATTWGT(  1,   0,   0)
                                        ATTRITION WEIGHT FOR BLUE SHIPS CROSSING RED SUBMARINE BARRIERS.
                                        (VALUE SHOULD BE BETWEEN 0.0 AND 1.0, INCLUSIVE.)
         .5000

VARIABLE ---- AVAILE(  5,   2,   0)
                                        (L,I) FRACTION OF THE RED ESCORTS THAT ARE AVAILABLE TO FLY MISSIONS
                                        FROM RED AIRBASE I TO TASK-FORCE LOCATION L DURING A TIME PERIOD IN
                                        WHICH AN ATTACK IS PLANNED.  THUS, THE OVERALL SORTIE RATE FOR ESCORTS
                                        IS GIVEN BY (1/IATKRT(L))*AVAILE(L,I) --
                                        I=1 IS THE VULNERABLE AIRBASE, I=2 IS THE INVULNERABLE AIRBASE.
        0.            0.
         .1000        0.
         .2000        0.
         .4000        0.
         .8000         .8000
```

VARIABLE ---- AVAILT(5, 2, 3)

(L,I,K) FRACTION OF RED BOMBERS OF TYPE K THAT ARE AVAILABLE TO FLY MISSIONS FROM RED AIRBASE I TO TASK-FORCE LOCATION L DURING A TIME PERIOD IN WHICH AN ATTACK IS PLANNED. THUS, THE OVERALL SORTIE RATE FOR TYPE-K BOMBERS IS GIVEN BY (1/IATKRT(L))*AVAILT(L,I,K) -- I=1 IS THE VULNERABLE AIRBASE, I=2 IS THE INVULNERABLE AIRBASE.

K = 1

0.	0.
.5000	0.
.5000	0.
.5000	.5000
.8000	.8000

K = 2

0.	0.
.5000	0.
.5000	0.
.5000	.5000
.8000	.8000

K = 3

0.	0.
0.	0.
0.	0.
.8000	.5000
.9000	.8000

VARIABLE ---- AVALED(5, 2, 0)

(L,IATF) AVAILABILITY FACTOR FOR RED ESCORT AIRCRAFT PERFORMING AIR-BASE DEFENSE WHEN THE (BLUE) TASK FORCE IS IN LOCATION L AND IATF=1--RED WILL ATTACK TASK FORCE LATER IN THE DAY IATF=2--RED WILL NOT.

.1000	.8000
.1100	.8100
.1200	.8200
.1300	.8300
.1400	.8400

VARIABLE ---- AWRCBB(1, 0, 0)

ATTRITION WEIGHT FOR RED SHIPS CROSSING BLUE SUBMARINE BARRIERS. (VALUE SHOULD BE BETWEEN 0.0 AND 1.0, INCLUSIVE.)

.5000

VARIABLE ---- BACCDW(6, 0, 0)

(KBS) COUNTERDETECTION WIDTH (DIAMETER) OF ASW AIRCRAFT FROM A BLUE SHIP OF KIND KBS AGAINST RED BARRIER SUBMARINES. (NMI)
KBS=1 IS FOR XPLAT, KBS=2 IS FOR XEAAW, KBS=3 IS FOR XEASWA,
KBS=4 IS FOR XEASWN, KBS=5 IS FOR XURGS, KBS=6 IS FOR BSSNDS.

0.	0.	20.00	0.	0.	0.

```
VARIABLE ---- BACPCK(  6,  0,  0)
                                  (KBS) CONDITIONAL PROBABILITY OF COUNTERKILL GIVEN DETECTION OF A
                                  RED BARRIER SUBMARINE BY A BLUE SHIP OF TYPE KBS.
                                  SEE BACCDW FOR DEFINITION OF KBS.
         .3000       0.             .2000       0.          0.          0.

VARIABLE ---- BAREAQ(  5,  0,  0)
                                  (L) THE AREA OF THE OUTER ASW SCREEN THAT CAN BE COVERED BY A SINGLE
                                  SEA-BASED ASW AIRCRAFT STATION IN LOCATION L. (NMI**2)
         1900.       1900.          1900.       1900.       1900.

VARIABLE ---- BARELQ(  5,  0,  0)
                                  (L) THE AREA OF THE OUTER ASW SCREEN THAT CAN BE COVERED BY A SINGLE
                                  LAND-BASED ASW AIRCRAFT STATION IN LOCATION L. (NMI**2)
         1900.       1900.          1900.       1000.       400.0

VARIABLE ---- BARLQ (  5,  0,  0)
                                  (L) THE LENGTH, OR CIRCUMFERENCE, OF THE OUTER ASW SCREEN IN LOCATION
                                  L. (NMI)
         500.0       500.0          1000.       1000.       1000.

VARIABLE ---- BARLTH(  5,  0,  0)
                                  (IBAR) LENGTH OF BARRIER BETWEEN REGIONS IBAR-1 AND IBAR. (NMI)
                                  BARLTH(IBAR) SHOULD NOT BE ZERO UNLESS ICTL(IBAR) IS ZERO. (SEE ICTL)
         10.00       200.0          100.0       50.00       150.0

VARIABLE ---- BECDW (  6,  0,  0)
                                  (KBS) COUNTERDETECTION WIDTH OF BLUE SHIP OF KIND KBS AGAINST RED BAR-
                                  RIER SUBMARINES. (NMI)
                                     SEE BACCDW FOR DEFINITION OF KBS.
         9.000       5.000          8.000       8.000       2.000       8.000

VARIABLE ---- BEDW   ( 10,  0,  0)
                                  (KRS) EFFECTIVE DETECTION WIDTH (DIAMETER) OF A BLUE BARRIER SUBMARINE
                                  AGAINST RED PENETRATING SHIPS OF KIND KRS. (NMI)
         10.00       10.00          15.00       20.00       0.          0.          0.          0.          0.          0.

VARIABLE ---- BMTMIN(  5,  0,  0)
                                  (L) MINIMUM NUMBER OF RED BOMBERS (TOTALED OVER ALL TYPES) REQUIRED
                                  TO LAUNCH AN ATTACK AGAINST THE TASK FORCE WHEN THE TASK FORCE IS IN
                                  LOCATION L.
         40.00       30.00          20.00       10.00       0.

VARIABLE ---- BSIBAR(  5,  0,  0)
                                  (IBAR) NUMBER OF BLUE SUBMARINES IN BARRIER BETWEEN REGIONS IBAR-1
                                  AND IBAR.
         0.          10.00          10.00       0.          0.
```

```
VARIABLE ---- BSSNDS(  1,   0,   0)
                                      NUMBER OF BLUE DIRECT-SUPPORT SUBMARINES.
          4.000

VARIABLE ---- BUCAP (  1,   0,   0)
                                      THE NUMBER OF SEA-BASED AIRCRAFT REQUIRED TO SUPPORT ONE CAP STATION.
          6.000

VARIABLE ---- CACDWO(  1,   0,   0)
                                      INITIAL CARRIER ASW AIRCRAFT COUNTERDETECTION WIDTH AGAINST RED BAR-
                                      RIER SUBMARINES. (NMI)
          35.00

VARIABLE ---- CAPMLQ(  5,   0,   0)
                                      (L) THE NUMBER OF CAP STATIONS ASSIGNED TO DEFEND EACH LAND-BASED AEW
                                      STATION WHEN THE TASK FORCE IS IN LOCATION L.
         0.         0.                 0.         1.000      1.500

VARIABLE ---- CAPMQ (  5,   0,   0)
                                      (L) THE NUMBER OF CAP STATIONS ASSIGNED TO DEFEND EACH SEA-BASED AEW
                                      STATION WHEN THE TASK FORCE IS IN LOCATION L.
          1.000      1.000              1.000      1.500      2.000

VARIABLE ---- CAPMR (  1,   0,   0)
                                      THE RANGE OF THE AIR-TO-AIR MISSILES CARRIED BY CAP AIRCRAFT. (NMI)
          50.00

VARIABLE ---- CAPSTQ(  5,   0,   0)
                                      (L) THE CAP STATION RADIUS FROM THE CENTER OF THE TASK FORCE IN
                                      LOCATION L. (NMI)
          400.0      300.0              200.0      100.0      100.0

VARIABLE ---- CPAGV (  1,   0,   0)
                                      CONDITIONAL PROBABILITY OF RED BARRIER ATTACK ON A CARRIER, GIVEN THAT
                                      CARRIER IS VULNERABLE TO DETECTION.
          .5000

VARIABLE ---- CPBPK (  6,   0,   0)
                                      (KBS) CONDITIONAL PROBABILITY THAT BLUE PENETRATING SHIP OF KIND KBS
                                      IS KILLED (GIVEN DETECTION BY RED BARRIER).
                                         SEE BACCDW FOR DEFINITION OF KBS.
          .8000      .8000              .8000      .8000      .8000      .7000

VARIABLE ---- CPBSCK( 10,   0,   0)
                                      (KRS) CONDITIONAL PROBABILITY THAT A BLUE BARRIER SUBMARINE, DETECTED
                                      BY A RED PENETRATING SHIP OF KIND KRS, IS COUNTERKILLED.
          .3000      .3000              .2000      .2000      0.         0.         0.         0.         0.         0.
```

VARIABLE ---- CPRPK (10, 0, 0)

(KRS) CONDITIONAL PROBABILITY THAT A RED PENETRATING SHIP OF KIND KRS IS KILLED.

.4500	.4500	.7500	.7500	0.	0.	0.	0.	0.	0.

VARIABLE ---- CPRSCK(6, 0, 0)

(KBS) CONDITIONAL PROBABILITY THAT A RED BARRIER SUBMARINE IS COUNTER-KILLED BY A BLUE SHIP OF TYPE KBS.
SEE BACCDW FOR DEFINITION OF KBS.

.1000	.1000	.1500	.1500	0.	.2000

VARIABLE ---- CSCDWO(1, 0, 0)

INITIAL CARRIER SONAR COUNTERDETECTION WIDTH AGAINST RED BARRIER SUB-MARINES. (NMI)

10.00

VARIABLE ---- DDFAC (10, 0, 0)

(KRS) FRACTION OF RED SHIPS OF KIND KRS CONDUCTING THE DDAY ATTACK THAT ATTACK CARRIERS.
NOTEINPUT VARIABLES STARTING WITH THE LETTERS DD OR IDD ARE USED IN SUBROUTINE DDAY.

.7000	.6000	.5000	.4000	0.	0.	0.	0.	0.	0.

VARIABLE ---- DDPKC (10, 0, 0)

(KRS) PROBABILITY OF KILL FOR A SHOT FROM A RED SHIP OF KIND KRS AGAINST A CARRIER ON DDAY.

.9000E-01	.6000E-01	.5000E-01	.4000E-01	0.	0.	0.	0.	0.	0.

VARIABLE ---- DDPKS (10, 0, 0)

(KRS) PROBABILITY OF KILL FOR A SHOT FROM A RED SHIP OF KIND KRS AGAINST A BLUE SHIP THAT IS NOT A CARRIER (REGARDLESS OF KIND OF BLUE SHIP) ON DDAY.

.1400	.1300	.8000E-01	.7000E-01	0.	0.	0.	0.	0.	0.

VARIABLE ---- DDRKAA(10, 0, 0)

(KRS) NUMBER OF RED SHIPS OF KIND KRS KILLED AFTER THE RED DDAY ATTACK ON BLUE.

1.000	1.000	1.000	1.000	0.	0.	0.	0.	0.	0.

VARIABLE ---- DDRKBA(10, 0, 0)

(KRS) NUMBER OF RED SHIPS OF KIND KRS KILLED BEFORE THE RED DDAY ATTACK ON BLUE.

1.000	1.000	1.000	1.000	0.	0.	0.	0.	0.	0.

VARIABLE ---- DDRSA (10, 0, 0)

(KRS) NUMBER OF RED SHIPS OF KIND KRS THAT WILL (IF ALIVE) ATTACK BLUE ON DDAY.

2.000	1.000	1.000	2.000	0.	0.	0.	0.	0.	0.

VARIABLE ---- DDSPA (10, 0, 0)

(KRS) SHOTS PER ATTACKING RED SHIP OF KIND KRS ON DDAY.

1.500	2.000	2.000	1.000	0.	0.	0.	0.	0.	0.

```
VARIABLE ---- DLIA   (  1,   0,   0)
                                        FIGHTER AVAILABILITY FACTOR FOR DLI.
                                        SEE IDA REPORT R-245 FOR ADDITIONAL INFORMATION.
         .6700

VARIABLE ---- D1T    (  2,   3,   0)
                                        (I,K) I=1 GIVES THE DISTANCE FROM TASK FORCE CENTER TO THE ASM RELEASE
                                        LINE OF RED BOMBERS OF TYPE K. (NMI)
                                        I=2 GIVES THE AVERAGE DISTANCE FROM TASK FORCE CENTER TO THE
                                        POINT AT WHICH RED BOMBERS OF TYPE K TURN
                                        AROUND--AFTER REACHING THIS POINT ATTACKING AIRCRAFT ARE INVULNERABLE.
         400.0        250.0               200.0
         200.0        200.0               100.0

VARIABLE ---- D2T    (  2,   3,   0)
                                        (I,K) I=1 GIVES THE AVERAGE DISTANCE FROM CARRIERS LAUNCHING DLIS TO
                                        THE TYPE-K RED BOMBERS ASM RELEASE LINE. (NMI)
                                        I=2 GIVES THE AVERAGE DISTANCE FROM CARRIERS LAUNCHING DLIS TO THE
                                        POINT AT WHICH RED BOMBERS OF TYPE K TURN
                                        AROUND--AFTER REACHING THIS POINT ATTACKING AIRCRAFT ARE INVULNERABLE.
         400.0        250.0               200.0
         200.0        200.0               100.0

VARIABLE ---- ENACDS( 10,   0,   0)
                                        (KRS) EXPECTED NUMBER OF AIRCRAFT (FIGHTERS AND ATTACKERS) DESTROYED
                                        WHEN A SHOT FROM A RED SURFACE SHIP OF TYPE KRS HITS A FULL CARRIER.
                                        NOTE-- KRS=1,2 ARE SUBMARINES, AND SO ENTRIES FOR ENACDS(1) AND
                                        ENACDS(2) ARE NOT USED.
        0.           0.                   1.500      .5000      0.         0.         0.         0.         0.         0.

VARIABLE ---- ENACDT(  4,   0,   0)
                                        (K) EXPECTED NUMBER OF AIRCRAFT (FIGHTERS AND ATTACKERS) DESTROYED
                                        WHEN AN ASM OF TYPE K HITS A FULL CARRIER
                                        -- K=NKRB+1 IS FOR ASMS LAUNCED FROM RED SUBMARINES.
         2.000        1.000               .5000      0.

VARIABLE ---- ESLR   (  1,   0,   0)
                                        THE LETHAL RANGE OF ASW ESCORTS FIRING ANTISUBMARINE MISSILES OR
                                        TORPEDOES. (NMI)
         8.000

VARIABLE ---- ESRQ   (  5,   0,   0)
                                        (L) THE ASW ESCORT STATION RADIUS FROM THE CENTER OF THE TASK FORCE IN
                                        LOCATION L. (NMI)
         8.000        10.00               15.00      15.00      15.00

VARIABLE ---- FAACA  (  5,   0,   0)
                                        (L) FRACTION OF BLUE ATTACK AIRCRAFT THAT WILL PERFORM AIRBASE ATTACK
                                        WHEN THE TASK FORCE IS IN LOCATION L. (AIRCRAFT ON CARRIERS ONLY.)
         .5000        .5100               .5200      .5300      .5400
```

VARIABLE ---- FACOB (5, 2, 0)

(KRA,IATF) FRACTION OF RED AIRCRAFT OF KIND KRA ON THE VULNERABLE RED AIRBASE WHEN--- IATF=1--RED WILL ATTACK TASK FORCE LATER IN THE CLOCK TIME PERIOD,--- IATF=2--RED WILL NOT.
KRA=1,NKRB FOR BOMBERS, KRA=NKRB+1 FOR ESCORTS, KRA=NKRB+2 FOR INTCT

.5100	.7100
.5200	.7200
.5300	.7300
.7400	.7400
.7500	.7500

VARIABLE ---- FFACA (5, 0, 0)

(L) FRACTION OF BLUE FIGHTER AIRCRAFT THAT WILL PERFORM AIRBASE ATTACK WHEN THE TASK FORCE IS IN LOCATION L.

.4000 .4100 .4200 .4300 .4400

VARIABLE ---- FFACE (5, 0, 0)

(L) FRACTION OF BLUE FIGHTER AIRCRAFT THAT WILL PERFORM AIRBASE ATTACK ESCORT WHEN THE TASK FORCE IS IN LOCATION L.

.2000 .2100 .2200 .2300 .2400

VARIABLE ---- FHSK (2, 0, 0)

(I) FRACTION OF HIT RED AIRCRAFT SHELTERS THAT ARE DESTROYED.
I=1--HIT BY A BLUE ATTACK AIRCRAFT
I=2--HIT BY A BLUE FIGHTER AIRCRAFT DOING AIRBASE ATTACK

.2200 .3200

VARIABLE ---- FM3 (6, 0, 0)

(KBS) FRACTION OF SHIPS OF KIND KBS CROSSING BARRIER IN FIRST WAVE, IF BARRIER CROSSING PROTOCOL 3 IS USED.
SEE BACCDW FOR DEFINITION OF KBS.

0. 0. 1.000 1.000 .2000 .5000

VARIABLE ---- FPPL1 (1, 0, 0)

THE NUMBER OF INCOMING ASMS THAT CAN BE FIRED AT BY A CARRIER'S POINT DEFENSE SAMS.

7.000

VARIABLE ---- FPPL2 (1, 0, 0)

THE NUMBER OF INCOMING ASMS THAT CAN BE FIRED AT BY A CARRIER'S POINT DEFENSE GUNS.

1.000

VARIABLE ---- FSTAQ (5, 0, 0)

(L) FRACTION OF MISSLE SUBMARINES IN LOCATION L THAT FIRE AT THE TASK FORCE DURING A TIME PERIOD IN WHICH A SUBMARINE ATTACK IS PLANNED.

.1000 .5000 1.000 1.000 1.000

```
VARIABLE ---- FSTGAO(  5,   0,   0)
                                     (L) FRACTION OF TORPEDO SUBMARINES IN LOCATION L THAT FIRE AT THE TASK
                                     FORCE DURING A TIME PERIOD IN WHICH A SUBMARINE ATTACK IS PLANNED.
            .1000        .5000         1.000        1.000        1.000

VARIABLE ---- HRMAAW(  1,   0,   0)
                                     NUMBER OF MISSLE  HITS (ONLY) REQUIRED TO NEUTRALIZE AN AAW SHIP.
            1.000

VARIABLE ---- HRMASW(  1,   0,   0)
                                     NUMBER OF MISSLE  HITS (ONLY) REQUIRED TO NEUTRALIZE AN ASW SHIP.
            1.000

VARIABLE ---- HRMURG(  1,   0,   0)
                                     NUMBER OF MISSLE  HITS (ONLY) REQUIRED TO NEUTRALIZE AN URG SHIP.
            1.000

VARIABLE ---- HRTAAW(  1,   0,   0)
                                     NUMBER OF TORPEDO HITS (ONLY) REQUIRED TO NEUTRALIZE AN AAW SHIP.
            1.000

VARIABLE ---- HRTASW(  1,   0,   0)
                                     NUMBER OF TORPEDO HITS (ONLY) REQUIRED TO NEUTRALIZE AN ASW SHIP.
            1.000

VARIABLE ---- HRTURG(  1,   0,   0)
                                     NUMBER OF TORPEDO HITS (ONLY) REQUIRED TO NEUTRALIZE AN URG SHIP.
            1.000

VARIABLE ---- IAADA (  1,   0,   0)
                                     ATTRITION  METHOD  USED FOR COMPUTING KILLS IN THE BLUE ATTACKER/RED
                                     DEFENDER AIR-TO-AIR INTERACTION IN SUBROUTINE ABATCK.  (THIS VARIABLE
                                     IS NOT CURRENTLY USED.)
                 0

VARIABLE ---- IAAED (  1,   0,   0)
                                     ATTRITION  METHOD  FOR COMPUTING KILLS IN THE BLUE ESCORT/RED DEFENDER
                                     INTERACTION IN ABATCK.   =0 IF ESCORTS CONCENTRATE FIRE,
                                                              =1 IF ESCORTS CONSERVE FIRE.
                 0

VARIABLE ---- IABAEQ(  1,   0,   0)
                                     INDEX OF ATTRITION METHOD USED TO COMPUTE RED LOSSES FROM AIRBASE
                                     ATTACK BY BLUE (SUBROUTINES ABATCK AND ATRTAB)
                                     IABAEQ=1--SHELTERS ATTACKED ONLY IF NO OPEN AIRCRAFT ARE DETECTED
                                           =2--ATTACKERS OPTIMALLY ALLOCATED TO SHELTERED VS OPEN AIRCRAFT
                                           =3--SHELTERS AND OPEN AIRCRAFT ARE ON SAME PARKING AREAS
                 1
```

```
VARIABLE ---- IABAF (  1,  0,  0)
                                     ATTRITION  METHOD  USED FOR COMPUTING KILLS IN THE BLUE AIRBASE ATTAC-
                                     KER/RED SAM INTERACTION.  (THIS VARIABLE IS NOT CURRENTLY USED.)
             0

VARIABLE ---- IABAW (  1,  0,  0)
                                     ATTRITION WEIGHTING SCHEME USED IN THE BLUE AIRBASE ATTACKER/RED SAM
                                     INTERACTION. IABAW=1--WEIGHTING BY TOTAL NUMBER OF SHOTS BY SAMS.
                                     (SEE CODE.)  IABAW=2--WEIGHTING BY NUMBER OF SAM FIRE CONTROL CENTERS.
             1

VARIABLE ---- IATKRT(  5,  0,  0)
                                     (L) INVERSE OF THE ATTACK RATE WHEN THE TASK FORCE IS IN LOCATION L.
                                     E.G., IATKRT(L)=1 MEANS RED ATTACKS EACH TIME PERIOD,
                                           IATKRT(L)=2 MEANS RED ATTACKS EVERY OTHER TIME PERIOD,
                                           IATKRT(L)=3 MEANS RED ATTACKS EVERY THIRD TIME PERIOD, ETC.
          1000          4                    3          2          1

VARIABLE ---- IATRIA(  1,  0,  0)
                                     INDEX FOR ATTRITION FUNCTION TO BE USED IN SUBROUTINE ATRTIA --
                                          =1 FOR INDEPENDENT TARGETING,
                                          =2 FOR COORDINATED TARGETING.
             1

VARIABLE ---- ICTL  (  5,  0,  0)
                                     (IBAR) INDICATOR FOR CONTROL OF BARRIER BETWEEN REGIONS IBAR-1 AND
                                     IBAR. 0--NO BARRIER PRESENT
                                           1--BLUE CONTROLS BARRIER (THEN RSIBAR(IBAR) IS EFFECTIVELY ZERO)
                                           2--RED CONTROLS BARRIER  (THEN BSIBAR(IBAR) IS EFFECTIVELY ZERO)
                                           3--BOTH BLUE AND RED HAVE BARRIERS
             0          1                    3          0          2

VARIABLE ---- IDDAC (  1,  0,  0)
                                     ATTRITION EQUATION USED TO COMPUTE ATTRITION TO BLUE CARRIERS ON DDAY.
                                       IDDAC=1--NO COORDINATION OF RED FIRE.
                                       IDDAC=2--PERFECT COORDINATION OF RED FIRE.
             2

VARIABLE ---- IDDAS (  1,  0,  0)
                                     ATTRITION EQUATION USED TO COMPUTE ATTRITION TO BLUE NON-CARRIER SHIPS
                                       IDDAS=1--NO COORDINATION OF RED FIRE.
                                       IDDAS=2--PERFECT COORDINATION OF RED FIRE.
                                     USED IN DDAY CALCULATIONS.
             1

VARIABLE ---- IKRAS (  5,  0,  0)
                                     (KRA) 1 IF RED AIRCRAFT OF KIND KRA CAN FIT INTO SHELTERS
                                           0 IF THEY CANNOT
                                     KRA=1,NKRB FOR BOMBERS,  KRA=NKRB+1 FOR ESCORTS,  KRA=NKRB+2 FOR INTCT
             0          1                    0          1          1
```

```
VARIABLE ---- IPLADA(  1,   0,   0)
                                    INDEX FOR ATTRITION  METHOD  USED IN DEFENDERS VS ATTACKERS
                                    INTERACTIONS WHEN RED PENETRATES THE BLUE LAND-BASED AIR BARRIER.
                                    (THIS INPUT IS NOT CURRENTLY USED.)
              0
VARIABLE ---- IPLAED(  1,   0,   0)
                                    INDEX FOR ATTRITION  METHOD  USED IN ESCORTS VS DEFENDERS
                                    INTERACTIONS WHEN RED PENETRATES THE BLUE LAND-BASED AIR BARRIER.
                                    =0 IF ESCORTS DO NOT CONSERVE FIRE FOR LATER INTERACTIONS,
                                    =1 IF ESCORTS CONSERVE FIRE FOR LATER INTERACTIONS.
              1
VARIABLE ---- IPPAF (  1,   0,   0)
                                    INDEX FOR ATTRITION FUNCTION FOR RED SAM/BLUE AIRCRAFT INTERACTIONS
                                    ON POWER PROJECTION MISSIONS. (NOT CURRENTLY USED)
              0
VARIABLE ---- IPPAW (  1,   0,   0)
                                    INDEX FOR ATTRITION WEIGHTING METHOD USED IN RED SAM/BLUE AIRCRAFT
                                    INTERACTIONS ON POWER PROJECTION MISSIONS.
                                      IPPAW=1 -- WEIGHT BY TOTAL NUMBER OF SHOTS BY SAMS.
                                      IPPAW=2 -- WEIGHT BY NUMBER OF SAM FIRE CONTROL CENTERS.
              2
VARIABLE ---- IRSUBA(  5,   0,   0)
                                    (L) INDEX GIVING RESTRICTION ON WHEN RED SUBS CAN ATTACK THE TASK
                                    FORCE GIVEN THAT THE TASK FORCE IS IN LOCATION L --
                                    0 = NO RESTRICTION, ALL SUBS CAN ATTACK EVERY TIME PERIOD
                                    1 = TORPEDO SUBS CAN ATTACK ALWAYS, MISSLE SUBS ATTACK ONLY WITH AIR
                                    2 = ALL SUBS ATTACK ONLY WHEN RED AIRCRAFT ALSO ATTACK THE TASK FORCE
              0          1          1          2          2
VARIABLE ---- ISSBR (  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, INDICATOR FOR DETECTION AND ATTACK PROTOCOL FOR
                                    BLUE SURFACE SHIPS ATTACKING RED  SURFACE SHIPS.
                                    ISSBR=0--DIFFERENT BLUE SHIPS DETECT INDEPENDENTLY OF ONE ANOTHER.
                                    ISSBR=1--TASK FORCE DETECTS AS AN INTEGRATED UNIT.
                                    (SEE ALSO THE DEFINITIONS OF VARIABLES SSPBDR AND SSPRDB, BELOW.)
              1
VARIABLE ---- ISSRB (  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, INDICATOR FOR DETECTION AND ATTACK PROTOCOL FOR
                                    RED  SURFACE SHIPS ATTACKING BLUE SURFACE SHIPS.
                                    ISSRB=0--A RED SHIP DETECTS DIFFERENT BLUE SHIPS INDEPENDENTLY.
                                    ISSRB=1--A RED SHIP DETECTS EITHER THE ENTIRE TASK FORCE OR NOTHING.
                                    (SEE ALSO THE DEFINITIONS OF VARIABLES SSPBDR AND SSPRDB, BELOW.)
              0
```

```
VARIABLE ---- LGTHMP(  6,   0,   0)
                                         (I) LEGNTH (IN TERMS OF NUMBER OF TIME PERIODS) OF MOVEMENT PERIOD I.
               1          1          1          2          2       1000
VARIABLE ---- LTFMP (  6,   0,   0)
                                         (I) LOCATION OF THE TASK FORCE DURING MOVEMENT PERIOD I.
               1          1          2          3          4          5
VARIABLE ---- MAXTP (  1,   0,   0)
                                         MAXIMUM NUMBER OF TIME PERIODS TO BE SIMULATED.
              30
VARIABLE ---- MIMP  (  1,   0,   0)
                                         MAXIMUM NUMBER OF MOVEMENT PERIODS TO BE SIMULATED.
               6
VARIABLE ---- NABSAM(  1,   0,   0)
                                         NUMBER OF KINDS OF RED SAM DEFENDING THE VULNERABLE RED AIRBASE.
               2
VARIABLE ---- NEPD  (  1,   0,   0)
                                         DUMMY VARIABLE USED ONLY BY SUBROUTINE INP.
               0
VARIABLE ---- NKBDPL(  1,   0,   0)
                                         NUMBER OF KINDS (TYPES) OF BLUE DEFENDING AIRCRAFT THAT FORM THE BLUE
                                         LAND-BASED AIR BARRIER.
               2
VARIABLE ---- NKRB  (  1,   0,   0)
                                         NUMBER OF DIFFERENT KINDS (TYPES) OF RED BOMBERS.
               3
VARIABLE ---- NKRS  (  1,   0,   0)
                                         NUMBER OF KINDS OF RED SHIP.  KINDS 1 AND 2 ARE TORPEDO SUBMARINES AND
                                         CRUISE MISSILE SUBMARINES, RESPECTIVELY.
               4
VARIABLE ---- NLOC  (  1,   0,   0)
                                         NUMBER OF POSSIBLE REGIONS (EXCLUDING REGION ZERO) IN WHICH THE TASK
                                         FORCE CAN BE LOCATED.
               5
VARIABLE ---- NPPSAM(  1,   0,   0)
                                         NUMBER OF TYPES OF RED SAMS INVOLVED IN BLUE POWER PROJECTION
                                         INTERACTIONS.
               2
```

VARIABLE ---- PAFCNF(1, 0, 0)

PROBABILITY THAT AN ATTACK AIRCRAFT, WHICH HAS ALREADY FLOWN A SORTIE DURING A CLOCK TIME PERIOD, CANNOT FLY ANOTHER SORTIE DURING THAT CLOCK TIME PERIOD.

1.000

VARIABLE ---- PARK (1, 0, 0)

NUMBER OF PARKING AREAS FOR RED AIRCRAFT ON A TYPICAL ACTUAL AIRBASE.

3.000

VARIABLE ---- PASS (2, 0, 0)

(I) NUMBER OF GROUND TARGETS THAT CAN BE ENGAGED ON AN ABA SORTIE.
I=1--BY A BLUE ATTACK AIRCRAFT
I=2--BY A BLUE FIGHTER AIRCRAFT DOING AIRBASE ATTACK

2.000 2.500

VARIABLE ---- PBDRN (2, 0, 0)

(I) PROBABILITY A BLUE AIRCRAFT ON AIRBASE ATTACK DETECTS A RED NON-SHELTERED AIRCRAFT. I=1--BLUE ATTACK AIRCRAFT
I=2--BLUE FIGHTER AIRCRAFT DOING AIRBASE ATTACK

.8700 .8700

VARIABLE ---- PBDRS (2, 0, 0)

(I) PROBABILITY A BLUE AIRCRAFT ON AIRBASE ATTACK DETECTS A RED SHELTER.
SEE PBDRN FOR THE DEFINITION OF I.

.4700 .4700

VARIABLE ---- PBKRN (2, 0, 0)

(I) PROBABILITY A BLUE AIRCRAFT ON AIRBASE ATTACK KILLS A RED NON-SHELTERED AIRCRAFT (GIVEN DETECTION).
SEE PBDRN FOR THE DEFINITION OF I.

.7600 .7700

VARIABLE ---- PBKRS (2, 0, 0)

(I) PROBABILITY A BLUE AIRCRAFT ON AIRBASE ATTACK KILLS A RED SHELTER.
SEE PBDRN FOR THE DEFINITION OF I.

.2600 .2700

VARIABLE ---- PDIN (1, 0, 0)

THE PROBABILITY THAT A SUBMARINE IS DETECTED BY THE ASW ESCORTS.
SEE IDA REPORT R-245 FOR ADDITIONAL INFORMATION.

.2000

VARIABLE ---- PFFCNF(1, 0, 0)

PROBABILITY THAT A FIGHTER AIRCRAFT, WHICH HAS ALREADY FLOWN A SORTIE DURING A CLOCK TIME PERIOD, CANNOT FLY ANOTHER SORTIE DURING THAT CLOCK TIME PERIOD.

.5000

```
VARIABLE ---- PKASW (  1,  0,  0)
                                THE PROBABILITY OF KILL, GIVEN DETECTION, OF A SUBMARINE (BOTH MISSILE
                                AND TORPEDO) BY AN ASW AIRCRAFT ON THE OUTER SCREEN. NO DISTINCTION
                                IS MADE BETWEEN LAND- OR SEA-BASED ASSETS.
          .5000
VARIABLE ---- PKAT1 (  1,  0,  0)
                                PROBABILITY OF KILL OF AN ATTACKING BOMBER OR ESCORT BY A SALVO OF
                                AIR-TO-AIR MISSILES FIRED FROM TASK FORCE AIRCRAFT.
          .8000
VARIABLE ---- PKDF1 (  1,  0,  0)
                                PROBABILITY OF KILL OF A DEFENDING FIGHTER (CAP OR DLI) BY A SALVO OF
                                AIR-TO-AIR MISSILES FIRED FROM RED ESCORT AIRCRAFT.
          .7500
VARIABLE ---- PKIIN (  1,  0,  0)
                                THE KILL PROBABILITY BY ASW MISSILE OR TORPEDO OF A SUBMARINE THAT
                                COMES WITHIN ESLR OF AN ASW ESCORT.
          .1500
VARIABLE ---- PKIN  (  1,  0,  0)
                                THE PROBABILITY THAT A SUBMARINE PROSECUTED BY AN AIRCRAFT FROM AN
                                AIR-CAPABLE ASW ESCORT WILL BE DESTROYED.
          .5000
VARIABLE ---- PKPLDT(  4,  0,  0)
                                (K) THE FRACTION OF INCOMING ASMS LAUNCHED FROM TYPE-K RED BOMBERS
                                THAT ARE EITHER DEFEATED BY A CARRIERS PASSIVE DEFENSE SYSTEMS OR ARE
                                UNRELIABLE FOR K=1,NKRB
                                 -- K=NKRB+1 IS FOR ASMS LAUNCHED FROM RED SUBMARINES.
          .5000       .5000       .5000       .5000
VARIABLE ---- PKPL1 (  1,  0,  0)
                                THE SINGLE SALVO PROBABILITY OF KILL OF INCOMING ASMS BY POINT DEFENSE
                                SAMS.
          .7500
VARIABLE ---- PKPL2 (  1,  0,  0)
                                THE SINGLE SALVO PROBABILITY OF KILL OF INCOMING ASMS BY POINT DEFENSE
                                GUNS.
          .3300
VARIABLE ---- PKSST (  4,  0,  0)
                                (K) THE SINGLE SALVO PROBABILITY OF KILL BY AAW ESCORTS OF INCOMING
                                ASMS LAUNCHED FROM TYPE-K RED BOMBERS FOR K=1,NKRB
                                 -- K=NKRB+1 IS FOR ASMS LAUNCHED FROM RED SUBMARINES.
          .7500       .7500       .7500       .7500
```

VARIABLE ---- PLAEDA(2, 0, 0)

(KBD) NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION TO 1.0) THAT A FRESH BLUE DEFENDER OF TYPE KBD CAN MAKE AGAINST RED ATTACKERS WHEN RED PENETRATES THE BLUE LAND-BASED AIR BARRIER.

3.000	0.

VARIABLE ---- PLAEDE(2, 0, 0)

(KBD) NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION TO 1.0) THAT A FRESH BLUE DEFENDER OF TYPE KBD CAN MAKE AGAINST RED ESCORTS WHEN RED PENETRATES THE BLUE LAND-BASED AIR BARRIER.

3.000	0.

VARIABLE ---- PLAEED(1, 0, 0)

NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION TO 1.0) THAT A FRESH RED ESCORT CAN MAKE AGAINST BLUE DEFENDERS WHEN RED PENETRATES THE BLUE LAND-BASED AIR BARRIER.

3.000

VARIABLE ---- PLBLBD(2, 5, 0)

(KBD,LB) NUMBER OF BLUE DEFENDING AIRCRAFT OF TYPE KBD ON LAND BASES ASSOCIATED WITH LOCATION LB.

0.	72.00	0.	0.	24.00
0.	50.00	0.	50.00	50.00

VARIABLE ---- PLCA (5, 0, 0)

(L) NUMBER OF COMBAT AREAS WHEN RED PENETRATES THE BLUE LAND-BASED AIR BARRIER AND THE TASK FORCE IS IN LOCATION L.

1.000	1.000	1.000	2.000	1.000

VARIABLE ---- PLFDLL(5, 5, 2)

(LB,L,K) FRACTION OF BLUE DEFENDING AIRCRAFT OF TYPE K WHOSE HOME AIRBASE IS ASSOCIATED WITH LOCATION LB THAT PRODUCE SORTIES THAT FORM THE LAND-BASED AIR BARRIER WHEN THE TASK FORCE IS IN LOCATION L.

K = 1

.5000	.6000	.2000	0.	0.
0.	.6000	.2000	0.	0.
0.	.1000	.3000	0.	0.
0.	0.	0.	.6000	.1000
0.	0.	0.	.3000	.2000

K = 2

.2000	.3000	0.	0.	0.
0.	.3000	.1000	0.	0.
0.	.2000	.2000	.1000	0.
0.	0.	0.	.3000	.2000
0.	0.	0.	.3000	.2000

VARIABLE ---- PLPAJO(3, 0, 0)

(KRA) PROBABILITY THAT A TYPE-KRA RED ATTACKER, WHEN ENGAGED BY A BLUE LAND-BASED DEFENDER, JETTISONS ITS ORDNANCE AND RETURNS FIRE (OTHERWISE IT ATTEMPTS TO OUTRUN THE DEFENDER).

.5000 .5000 .5000

VARIABLE ---- PLPDDA(2, 0, 0)

(KBD) PROBABILITY OF DETECTION BY A BLUE TYPE-KBD LAND-BASED DEFENDER OF A RED ATTACKER PENETRATING THE LAND-BASED AIR BARRIER.

1.000 1.000

VARIABLE ---- PLPDDE(2, 0, 0)

(KBD) PROBABILITY OF DETECTION BY A BLUE TYPE-KBD LAND-BASED DEFENDER OF A RED ESCORT ENETRATING THE LAND-BASED AIR BARRIER.

1.000 1.000

VARIABLE ---- PLPDED(1, 0, 0)

PROBABILITY OF DETECTION BY A RED ESCORT OF A BLUE DEFENDER WHEN RED IS PENETRATING THE BLUE LAND-BASED AIR BARRIER.

1.000

VARIABLE ---- PLPKAD(3, 2, 0)

(KRA,KBD) PROBABILITY THAT A RED TYPE-KRA ATTACKER KILLS A BLUE TYPE-KBD DEFENDER GIVEN THAT RED IS PENETRATING THE BLUE LAND-BASED AIR BARRIER AND THAT THE RED ATTACKER HAS BEEN ENGAGED BY THE BLUE DEFENDER AND HAS JETTISONED ITS (AIR-TO-GROUND) ORDNANCE.

0.	0.
0.	0.
.1000	.2000

VARIABLE ---- PLPKDA(2, 3, 0)

(KBD,KRA) PROBABILITY THAT A BLUE TYPE KBD-DEFENDER KILLS A RED TYPE-KRA ATTACKER GIVEN THAT RED IS PENETRATING THE BLUE LAND-BASED AIR BARRIER AND THAT THE BLUE DEFENDER IS ENGAGING THE RED ATTACKER.

.6000	.8000	.5000
.4000	.5000	.1000

VARIABLE ---- PLPKDE(2, 0, 0)

(KBD) PROBABILITY THAT A BLUE TYPE-KBD DEFENDER KILLS A RED ESCORT GIVEN THAT THEY ARE ENGAGED WHEN RED IS PENETRATING THE BLUE LAND-BASED AIR BARRIER.

.5000 .1000

VARIABLE ---- PLPKED(2, 0, 0)

(KBD) PROBABILITY THAT A RED ESCORT KILLS A BLUE TYPE-KBD DEFENDER GIVEN THAT THEY ARE ENGAGED WHEN RED IS PENETRATING THE BLUE LAND-BASED AIR BARRIER.

.1000 .2000

```
VARIABLE ---- PPAEGS(  2,   0,   0)
                                      (KBA) AVERAGE NUMBER OF ADDITIONAL GROUND TARGETS THAT A BLUE POWER
                                      PROJECTION AIRCRAFT OF TYPE KBA CAN ENGAGE AFTER IT HAS ENGAGED ONE
                                      RED SAM IN A SAM-SUPPRESSION ROLE.
        7.000        3.000

VARIABLE ---- PPANMS(  2,   0,   0)
                                      (KRS) ACTUAL NUMBER OF MISSLES AVAILABE FOR USE BY RED TYPE KRS SAMS
                                      WHICH ARE DEFENDING AGAINST BLUE POWER PROJECTION.
        1000.        100.0

VARIABLE ---- PPAVLS(  2,   5,   0)
                                      (KRS,L) AVAILABILITY FACTOR FOR RED SAMS OF TYPE KRS DEFENDING AGAINST
                                      BLUE POWER PROJECTION WHEN THE TASK FORCE IS IN LOCATION L.
        1.000        1.000             .2000        .2000        .2000
        1.000        1.000             .9000        .6000        .6000

VARIABLE ---- PPAVSS(  2,   0,   0)
                                      (KRS) AVERAGE NUMBER OF POSSIBLE SHOTS BY EACH TYPE-KRS RED SAM PER
                                      BLUE AIRCRAFT ON POWER PROJECTION MISSIONS AT THESE BLUE AIRCRAFT.
        .5000        1.000

VARIABLE ---- PPCAL (  5,   0,   0)
                                      (L) NUMBER OF COMBAT AREAS FOR BLUE-POWER-PROJECTION/RED-SAM
                                      INTERACTIONS WHEN THE TASK FORCE IS IN LOCATION L.
        1.000        1.000             1.000        1.000        1.000

VARIABLE ---- PPFASM(  2,   0,   0)
                                      (KBA) FRACTION OF THOSE BLUE TYPE-KBA AIRCRAFT ENGAGING RED SAMS
                                      ON POWER PROJECTION MISSIONS THAT USE STANDOFF MUNITIONS.
        .5000        0.

VARIABLE ---- PPFASS(  2,   0,   0)
                                      (KBA) FRACTION OF BLUE TYPE-KBA AIRCRAFT ON POWER PROJECTION MISSIONS
                                      THAT ATTEMPT TO DO SAM SUPPRESSION.
        1.000        .5000

VARIABLE ---- PPFSVS(  2,   0,   0)
                                      (KRS) FRACTION OF RED TYPE-KRS SAMS THAT ARE VULNERABLE TO BLUE
                                      SAM SUPPRESSORS ON POWER PROJECTION MISSIONS.
        .5000        1.000

VARIABLE ---- PPPDAS(  2,   0,   0)
                                      (KBA) PROBABILITY THAT A BLUE TYPE-KBA SAM SUPPRESSOR DETECTS ANY
                                      PARTICULAR RED SAM ON POWER PROJECTION MISSIONS.
        .1000E-01    1.000
```

```
VARIABLE ---- PPPDSA(  2,   0,   0)
                                          (KRS) PROBABILITY THAT A RED SAM DETECTS ANY PARTICULAR BLUE AIRCRAFT
                                          ON POWER PROJECTION MISSIONS.
           .1000        1.000

VARIABLE ---- PPPKAS(  2,   0,   0)
                                          (KRS) PROBABILITY THAT A RED SAM OF TYPE KRS THAT HAS BEEN SUPPRESSED
                                          BY ONE OR MORE BLUE SAM SUPPRESSORS IS KILLED BY THOSE SUPPRESSORS
                                          ON POWER PROJECTION MISSIONS.
           .8000        1.000

VARIABLE ---- PPPKSA(  2,   2,   0)
                                          (KRS,KBA) PROBABILITY THAT A RED SAM OF TYPE KRS KILLS A BLUE AIRCRAFT
                                          OF TYPE KBA ON POWER PROJECTION MISSIONS GIVEN THAT THE SAM IS
                                          SHOOTING AT THAT AIRCRAFT.
           .4000        .6000
           .6000        .4000

VARIABLE ---- PPPSAS(  2,   2,   0)
                                          (KBA,KRS) PROBABILITY THAT A BLUE SAM SUPPRESSOR OF TYPE KBA
                                          SUPPRESSES A RED TYPE-KRS SAM ON POWER PROJECTION MISSIONS GIVEN THAT
                                          THE AIRCRAFT IS SHOOTING AT THAT SAM.
           .4000        .5000
           .2000        .5000

VARIABLE ---- PPRSAM(  2,   0,   0)
                                          (KRS) NUMBER OF RED SAMS OF TYPE KRS THAT ARE PROVIDING DEFENSE
                                          AGAINST BLUE POWER PROJECTION MISSIONS.
           20.00        10.00

VARIABLE ---- PPSURR(  2,   5,   0)
                                          (KBA,L) SORTE RATE FOR BLUE AIRCRAFT OF TYPE KBA ON POWER PROJECTION
                                          MISSIONS WHEN THE TASK FORCE IS IN LOCATION L --
                                            KBA = 1  FOR ATTACK AIRCRAFT,
                                            KBA = 2  FOR FIGHTR AIRCRAFT.
          0.           0.                   .5000        1.000        2.000
          0.           0.                   .5000        1.000        2.000

VARIABLE ---- PPTSCS(  2,   0,   0)
                                          (KRS) TOTAL NUMBER OF SHOTS PER TIME PERIOD THAT A TYPE-KRS RED SAM
                                          CAN MAKE WHEN DEFENDING AGAINST BLUE POWER PROJECTION MISSIONS.
           2.000        4.000
```

```
VARIABLE ---- PRSM  ( 10,   5,   6)
                              (KRS,LOC,LOCTF1) PROPORTION OF RED SHIPS OF KIND KRS IN LOCATION LOC
                              MOVING TOWARD TASK FORCE WHEN TASK FORCE IS IN LOCATION LOCTF1-1.
                              RED SHIPS IN SAME LOCATION AS TASK FORCE DO NOT MOVE.
                              (DATA FOR PRSM(KRS,LOCTF1-1,LOCTF1) ARE IGNORED.)
K = 1
         .5800      .9500      .7800      .2900E-01   .4500E-01
        0.          .2700      .3000      .6800E-01   .3800E-01
         .1300      .8300      .5800      .9000E-02   .2700E-01
         .6200      .8000E-01  .9900      .9700E-01   .6900E-01
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
K = 2
        0.          .5800      .9000      .8800E-01   .4900E-01
        0.          .8700      .9300      .5900E-01   .2000E-01
        0.          .4600      .6200      .4400E-01   .4500E-01
        0.          .7600      .5900      .4400E-01   .4900E-01
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
K = 3
         .3700     0.          .3100      .6600E-01   .6500E-01
         .5100     0.          .3800      .1100E-01   .8500E-01
         .7100     0.          .6400      .1000E-02   .7000E-02
         .9500     0.          .3800      .6000E-01   .8900E-01
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
K = 4
         .6200      .8000     0.          .5100E-01   .6300E-01
         .7400      .7000     0.          .7400E-01   .5600E-01
         .3000      .4800     0.          .8000E-01   .4000E-02
         .8900      .8500     0.          .6800E-01   .5000E-01
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
        0.         0.         0.         0.          0.
```

```
K = 5
        .7300      .3600      .6400      0.          .8800E-01
        .3800      .7600      .8800      0.          .7000E-02
        .7600      .9100      .7600      0.          .5200E-01
        .9500      .9500      .9100      0.          .3200E-01
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
K = 6
        .7000      .4000E-01  .1800      .6000E-02  0.
        .7300      .5200      .4900      .1000E-01  0.
        .3000E-01  .1800      .9100      .6900E-01  0.
        .9400      .9500      .5900      .5200E-01  0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.
       0.         0.         0.         0.         0.

VARIABLE ---- PRWLNQ(  5,   0,   0)
                                   (L) PROBABILITY THAT LAND-BASED RESOURCES ARE ABLE TO WARN THE TASK
                                   FORCE (IN LOCATION L) OF AN IMPENDING RED AIR ATTACK.
        1.000      1.000      0.         .5000      0.

VARIABLE ---- RACCDW( 10,   0,   0)
                                   (KRS) COUNTERDETECTION WIDTH OF ASW AIRCRAFT FROM A RED SHIP OF KIND
                                   KRS AGAINST BLUE BARRIER SUBMARINES. (NMI)
       0.         0.          50.00      30.00      0.         0.         0.         0.         0.         0.

VARIABLE ---- RACPCK( 10,   0,   0)
                                   (KRS) CONDITIONAL PROBABILITY OF COUNTERKILL GIVEN DETECTION.
       0.         0.          .8000      .7500      0.         0.         0.         0.         0.         0.

VARIABLE ---- RARBAB(  3,   0,   0)
                                   (K) NUMBERS OF RED AIRCRAFT REQUIRED ON RED AIRBASES IN ORDER FOR BLUE
                                   TO ATTACK THE VULNERABLE RED AIRBASE. ALL CRITERIA MUST BE SATISFIED--
                                   K=1 IS NUMBER OF BOMBERS REQUIRED ON ALL RED AIRBASES,
                                   K=2 IS NUMBER OF BOMBERS AND FIGHTERS REQUIRED ON ALL RED AIRBASES,
                                   K=3 IS NUMBER OF BOMBERS AND FIGHTERS REQUIRED ON VULNERABLE RED BASE.
        10.00      15.00      5.000

VARIABLE ---- RECDW ( 10,   0,   0)
                                   (KRS) COUNTERDETECTION WIDTH OF RED SHIP OF KIND KRS AGAINST BLUE BAR-
                                   RIER SUBMARINES. (NMI)
        10.00      10.00      13.00      15.00      0.         0.         0.         0.         0.         0.
```

```
VARIABLE ---- REDW   (  6,   0,   0)
                                    (KBS) EFFECTIVE DETECTION WIDTH OF A RED BARRIER SUBMARINE AGAINST
                                    BLUE PENETRATING SHIPS OF KIND KBS. (NMI)
                                       SEE BACCDW FOR DEFINITION OF KBS.
        20.00        20.00           15.00        15.00        15.00        10.00

VARIABLE ---- PSIBAR(  5,   0,   0)
                                    (IBAR) NUMBER OF RED SUBMARINES IN BARRIER BETWEEN REGIONS IBAR-1
                                    AND IBAR.
       0.           0.               5.000       0.            5.000

VARIABLE ---- RS     ( 10,   5,   0)
                                    (KRS,LOC) NUMBER OF RED SHIPS OF KIND KRS IN LOCATION (REGION) LOC.
                                      KRS=1 FOR TORPEDO SUBMARINES
                                      KRS=2 FOR CRUISE MISSLE SUBMARINES
                                      KRS=3,NKRS FOR SURFACE SHIPS.
       0.            .5000           1.000        2.200        3.000
        3.000       0.                .7000       1.200        3.000
       0.           0.              0.            15.00        25.00
        2.000        8.000           11.00        5.000        20.00
       0.           0.              0.           0.           0.
       0.           0.              0.           0.           0.
       0.           0.              0.           0.           0.
       0.           0.              0.           0.           0.
       0.           0.              0.           0.           0.
       0.           0.              0.           0.           0.

VARIABLE ---- SBFBCF(  1,   0,   0)
                                    IN SUBROUTINE SUBSUB, FRACTION OF BLUE SSN(DS) INVOLVED IN THE COMBAT
                                    THAT ARE CAPABABLE OF SHOOTING AT RED SURFACE SHIPS.
                                    NOTE -- SSN(DS) ARE BSSNDS SUBMARINES.
        .8600

VARIABLE ---- SBFBCS(  1,   0,   0)
                                    IN SUBROUTINE SUBSUB, FRACTION OF BLUE SSN(DS) INVOLVED IN THE COMBAT
                                    THAT ARE CAPABABLE OF SHOOTING AT RED SUBMARINES.
        .8700

VARIABLE ---- SBFRFA(  5,   0,   0)
                                    (L) IN SUBROUTINE SUBSUB, FRACTION OF RED SURFACE SHIPS IN REGION L
                                    INVOLVED IN THE COMBAT (WHEN TASK FORCE IS IN REGION L).
        .8500        .8600           .8700        .8800        .8900

VARIABLE ---- SBFRFC(  1,   0,   0)
                                    IN SUBROUTINE SUBSUB, FRACTION OF RED SURFACE SHIPS INVOLVED IN THE
                                    COMBAT THAT ARE CAPABLE OF SHOOTING AT BLUE SSN(DS).
        .8900
```

```
VARIABLE ---- SBFRSA(  5,  0,  0)
                                        (L)IN SUBROUTINE SUBSUB, FRACTION OF RED SUBMARINES IN REGION L
                                        INVOLVED IN THE COMBAT (WHEN TASK FORCE IS IN REGION L).
              .9000        .9100          .9200        .9300        .9400

VARIABLE ---- SBFRSC(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, FRACTION OF RED SUBMARINES INVOLVED IN THE
                                        COMBAT THAT ARE CAPABLE OF SHOOTING AT BLUE SSN(DS).
              .9100

VARIABLE ---- SBPBDF(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN BLUE SSN(DS) DETECTS
                                        A GIVEN RED SURFACE SHIP.
              .6200

VARIABLE ---- SBPBDS(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN BLUE SSN(DS) DETECTS
                                        A GIVEN RED SUBMARINE.
              .7200

VARIABLE ---- SBPBKF(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN BLUE SSN(DS) KILLS (GIVEN
                                        DETECTION) A GIVEN RED SURFACE SHIP.
              .3600

VARIABLE ---- SBPBKS(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN BLUE SSN(DS) KILLS (GIVEN
                                        DETECTION) A GIVEN RED SUBMARINE.
              .4800

VARIABLE ---- SBPFDB(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN RED SURFACE SHIP DETECTS
                                        A GIVEN BLUE SUBMARINE.
              .7100

VARIABLE ---- SBPFKB(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN RED SURFACE SHIP KILLS
                                        (GIVEN DETECTION) A GIVEN BLUE SUBMARINE.
              .2300

VARIABLE ---- SBPSDB(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN RED SUBMARINE DETECTS
                                        A GIVEN BLUE SUBMARINE.
              .8100

VARIABLE ---- SBPSKB(  1,  0,  0)
                                        IN SUBROUTINE SUBSUB, PROBABILITY A GIVEN RED SUBMARINE KILLS
                                        (GIVEN DETECTION) A GIVEN BLUE SUBMARINE.
              .3300
```

VARIABLE ---- SHEL (1, 0, 0)
INITIAL NUMBER OF RED AIRCRAFT SHELTERS ON THE VULNERABLE RED AIRBASE.
100.0

VARIABLE ---- SMALLR(1, 0, 0)
THE RANGE OF THE AIR-TO-AIR MISSILES CARRIED BY THE DLI. (NMI)
50.00

VARIABLE ---- SSBACR(8, 0, 0)
(KRSS) NUMBER OF BLUE AIRCRAFT (ATTACK AIRCRAFT OR EQUIVALENT) FROM
CARRIER REQUIRED TO DESTROY A RED SURFACE SHIP OF KIND KRSS.
KRSS=KRS-2 VARIES FROM 1 TO NKRS-2, INDEXES KIND OF RED SURFACE SHIP,
CORRESPONDING TO KRS=3 TO NKRS. (KRS=1 AND 2 ARE RED SUBMARINES.)

50.00	10.00	0.	0.	0.	0.	0.	0.

VARIABLE ---- SSCFA (1, 0, 0)
ONE BLUE FIGHTER AIRCRAFT IS EQUIVALENT TO SSCFA BLUE ATTACK AIRCRAFT,
FOR PURPOSES OF DESTROYING RED SURFACE SHIPS. (SUBROUTINE SHPSHP)
.8000

VARIABLE ---- SSDAAW(1, 0, 0)
PROBABILITY OF KILL OF AN ASM BY THE SSD SYSTEMS ON AN AAW SHIP.
.9000

VARIABLE ---- SSDASW(1, 0, 0)
PROBABILITY OF KILL OF AN ASM BY THE SSD SYSTEMS ON AN ASW SHIP.
.1000

VARIABLE ---- SSDURG(1, 0, 0)
PROBABILITY OF KILL OF AN ASM BY THE SSD SYSTEMS ON AN URG SHIP.
0.

VARIABLE ---- SSFBAK(2, 8, 0)
(KBA,KRSS) FRACTION OF BLUE AIRCRAFT OF TYPE KBA KILLED ATTACKING RED
SURFACE SHIPS OF TYPE KRSS.
KRSS=KRS-2 VARIES FROM 1 TO NKRS-2, INDEXES KIND OF RED SURFACE SHIP,
CORRESPONDING TO KRS=3 TO NKRS. (KRS=1 AND 2 ARE RED SUBMARINES.)
KBA=1 FOR BLUE ATTACK AIRCRAFT, KBA=2 FOR BLUE FIGHTER AIRCRAFT.

.1000	.5000E-01	0.	0.	0.	0.	0.	0.
.2000	.1000	0.	0.	0.	0.	0.	0.

```
VARIABLE ---- SSFRSV(  8,   5,   0)
                                    (KRSS,L) FRACTION OF RED SURFACE SHIPS OF KIND KRSS THAT ARE VULNER-
                                    ABLE TO ATTACK BY BLUE TASK FORCE WHEN THE TASK FORCE IS IN REGION L.
                                    KRSS=KRS-2 VARIES FROM 1 TO NKRS-2, INDEXES KIND OF RED SURFACE SHIP,
                                    CORRESPONDING TO KRS=3 TO NKRS. (KRS=1 AND 2 ARE RED SUBMARINES.)
        1.000        1.000        .7500        .5000        .7500
        1.000        1.000        .9000        .8000        .7000
       0.           0.           0.           0.           0.
       0.           0.           0.           0.           0.
       0.           0.           0.           0.           0.
       0.           0.           0.           0.           0.
       0.           0.           0.           0.           0.
       0.           0.           0.           0.           0.

VARIABLE ---- SSPBDR(  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, PROBABILITY BLUE DETECTS RED.
                                    IF ISSBR=0, SSPBDR IS THE PROB. THAT A GIVEN BLUE SURFACE SHIP DETECTS
                                    A GIVEN RED  SURFACE SHIP.  IF ISSBR=1, SSPBDR IS THE PROB. THAT THE
                                    BLUE TASK FORCE (AS A UNIT) DETECTS A GIVEN RED SURFACE SHIP.
        .8000

VARIABLE ---- SSPBKR(  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, PROBABILITY A GIVEN BLUE SURFACE SHIP KILLS A
                                    GIVEN RED  SURFACE SHIP, GIVEN (DETECTION AND) ATTACK.
                                    NOTE--THE PROB. OF ATTACK IS GOVERNED BY VARIABLES ISSBR AND ISSRB.
        .3000

VARIABLE ---- SSPRDB(  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, PROBABILITY RED DETECTS BLUE.
                                    IF ISSRB=0, SSPRDB IS THE PROB. THAT A GIVEN RED  SURFACE SHIP DETECTS
                                    A GIVEN BLUE SURFACE SHIP.  IF ISSRB=1, SSPRDB IS THE PROB. THAT A
                                    GIVEN RED SURFACE SHIP DETECTS THE BLUE TASK FORCE.
        .5000

VARIABLE ---- SSPRKB(  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, PROBABILITY A GIVEN RED  SURFACE SHIP KILLS A
                                    GIVEN BLUE NON-CARRIER SURFACE SHIP, GIVEN (DETECTION AND) ATTACK.
                                    NOTE--THE PROB. OF ATTACK IS GOVERNED BY VARIABLES ISSBR AND ISSRB.
        .6500

VARIABLE ---- SSPRKC(  1,   0,   0)
                                    IN SUBROUTINE SHPSHP, PROBABILITY A GIVEN RED  SURFACE SHIP KILLS A
                                    GIVEN BLUE CARRIER, GIVEN  (DETECTION AND) ATTACK.
                                    NOTE--THE PROB. OF ATTACK IS GOVERNED BY VARIABLES ISSBR AND ISSRB.
                                    NOTE--CARRIERS ARE NOT ACTUALLY KILLED.  THEIR CAPABILITY IS DEGRADED.
        .1200
```

VARIABLE ---- STARQ (5, 0, 0)

(L) THE AEW STATION RADIUS FROM THE CENTER OF THE TASK FORCE IN LOCATION L. (NMI)

300.0	300.0	300.0	250.0	200.0

VARIABLE ---- STSALV(1, 0, 0)

THE NUMBER OF TORPEDO SALVOS FIRED BY EACH SURVIVING TORPEDO SUBMARINE.

2.000

VARIABLE ---- SUBSOR(1, 0, 0)

THE DISTANCE FROM TASK FORCE CENTER TO THE TORPEDO RELEASE LINE. (NMI)

15.00

VARIABLE ---- TAB10T(20, 4, 0)

(I,K) TAB10(I,K) IS THE NUMBER OF ASMS THAT CAN BE ENGAGED BY I AAW ESCORTS GIVEN THAT THESE ASMS WERE LAUNCED FROM TYPE-K RED BOMBERS FOR K=1,NKRB. FOR K=NKRB+1, TAB10 IS THIS NUMBER GIVEN THAT THE ASMS WERE LAUNCHED FROM RED SUBMARINES.

8.000	8.000	8.000	8.000
16.00	16.00	16.00	16.00
24.00	24.00	24.00	24.00
32.00	32.00	32.00	32.00
40.00	40.00	40.00	40.00
48.00	48.00	48.00	48.00
56.00	56.00	56.00	56.00
64.00	64.00	64.00	64.00
72.00	72.00	72.00	72.00
80.00	80.00	80.00	80.00
88.00	88.00	88.00	88.00
96.00	96.00	96.00	96.00
104.0	104.0	104.0	104.0
112.0	112.0	112.0	112.0
120.0	120.0	120.0	120.0
128.0	128.0	128.0	128.0
136.0	136.0	136.0	136.0
144.0	144.0	144.0	144.0
152.0	152.0	152.0	152.0
160.0	160.0	160.0	160.0

VARIABLE ---- TAB12 (20, 0, 0)

(I) A VECTOR IN WHICH THE ITH COMPONENT IS THE CARRIER SURVIVAL PROBABILITY IF IT RECEIVES I TORPEDO HITS.

.8000	.4000	.2000	.1000	.5000E-01	.2500E-01	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE ---- TAB13T(20, 4, 0)

(I,K) TAB13(I,K) IS THE CARRIER SURVIVAL PROBABILITY IF IT RECEIVES I HITS BY ASMS LAUNCED FROM TYPE-K RED BOMBERS FOR K=1,NKRB. FOR K=NKRB+1, TAB13 IS THIS PROBABILITY GIVEN THAT THE ASMS WERE LAUNCHED FROM RED SUBMARINES.

.6000	.6000	1.000	.6000
.3000	.3000	.8000	.3000
.1500	.1500	.6000	.1500
.7500E-01	.7500E-01	.4000	.7500E-01
.3500E-01	.3500E-01	.2000	.3500E-01
.1500E-01	.1500E-01	.1000	.1500E-01
.1000E-01	.1000E-01	.5000E-01	.1000E-01
.5000E-02	.5000E-02	.2500E-01	.5000E-02
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

VARIABLE ---- TCAP (1, 0, 0)

A TIME DELAY BETWEEN WHEN THE BOMBER RAID IS DETECTED AND WHEN THE CAP AIRCRAFT BEGIN TO FLY TOWARD ENGAGEMENT. (MIN)

1.500

VARIABLE ---- THSCAO(5, 0, 0)

(L) WIDTH OF THREAT SECTOR FOR ANTI-SHIP MISSLE ATTACKS IN LOCATION L. (DEG)

120.0	90.00	360.0	180.0	360.0

VARIABLE ---- THSCTO(5, 0, 0)

(L) WIDTH OF THREAT SECTOR FOR TORPEDO ATTACKS IN LOCATION L. (DEG)

90.00	60.00	360.0	180.0	360.0

VARIABLE ---- TPAS (1, 0, 0)

NUMBER OF TORPEDOES FIRED BY A RED BARRIER SUBMARINE IN AN ATTACK AGAINST A CARRIER.

2.500

VARIABLE ---- TPS (1, 0, 0)

THE NUMBER OF TORPEDOES PER TORPEDO SALVO.

2.500

```
VARIABLE ---- T1      ( 1, 0, 0)
                                  THE TIME BETWEEN DETECTION OF THE BOMBER RAID AND LAUNCH OF THE FIRST
                                  WAVE OF DLIS. (MIN)
          8.000

VARIABLE ---- T2      ( 1, 0, 0)
                                  A TIME PENALTY FOR DISTANCE MADE GOOD DURING DLI TAKE-OFF AND CLIMB.
                                  (MIN)
          3.000

VARIABLE ---- T3      ( 1, 0, 0)
                                  ANY MISCELLANEOUS TIME DELAY TO BE ASSIGNED TO THE DLI MISSION. (MIN)
        0.

VARIABLE ---- T4      ( 1, 0, 0)
                                  THE TIME BETWEEN DLI WAVES. (MIN)
          1.000

VARIABLE ---- UBAEWL( 1, 0, 0)
                                  THE RECIPROCAL OF THE NUMBER OF LAND-BASED AEW AIRCRAFT NEEDED TO
                                  SUPPORT ONE AEW STATION.
          .1400

VARIABLE ---- UBAEW ( 1, 0, 0)
                                  THE RECIPROCAL OF THE NUMBER OF SEA-BASED AEW AIRCRAFT NEEDED TO
                                  SUPPORT ONE AEW STATION.
          .1400

VARIABLE ---- UBASWL( 1, 0, 0)
                                  THE RECIPROCAL OF THE NUMBER OF LAND-BASED ASW AIRCRAFT REQUIRED TO
                                  SUPPORT A STATION ON THE OUTER ASW SCREEN.
          .1000

VARIABLE ---- UBASW ( 1, 0, 0)
                                  THE RECIPROCAL OF THE NUMBER OF SEA-BASED ASW AIRCRAFT REQUIRED TO
                                  SUPPORT A STATION ON THE OUTER ASW SCREEN.
          .2500

VARIABLE ---- VBT     ( 3, 0, 0)
                                  (K) THE VELOCITY OF RED BOMBERS OF TYPE K. (NMI/MIN)
          20.00       10.00         10.00

VARIABLE ---- VCAP    ( 1, 0, 0)
                                  THE VELOCITY OF CAP AIRCRAFT WHEN ENGAGING THE BOMBER RAID. (NMI/MIN)
          12.00

VARIABLE ---- VI      ( 1, 0, 0)
                                  THE AVERAGE VELOCITY OF DLIS FROM LAUNCH TO ENGAGEMENT. (NMI/MIN)
          12.00
```

VARIABLE ---- WFMAAW(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF MISSLES THAT ARE TARGETED AGAINST AAW SHIPS.

.2500E-01

VARIABLE ---- WFMASW(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF MISSLES THAT ARE TARGETED AGAINST ASW SHIPS.

.2500E-01

VARIABLE ---- WFMPLT(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF MISSLES THAT ARE TARGETED AGAINST AIRCRAFT CARRIERS.

.2500

VARIABLE ---- WFMURG(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF MISSLES THAT ARE TARGETED AGAINST URG SHIPS.

.2500E-01

VARIABLE ---- WFPPAS(2, 5, 0)
(KBA,L) WEIGHTING FACTOR APPLIED TO SUCCESSFUL POWER PROJECTION SORTIES FLOWN BY TYPE-KBA BLUE AIRCRAFT FROM LOCATION L TO DETERMINE CUMULATIVE WEIGHTED POWER PROJECTION SORTIES FOR SUMMARY DISPLAY,
KBA=1 FOR ATTACK AIRCRAFT, KBA=2 FOR FIGHTER AIRCRAFT.

0.	0.	.5000	1.000	1.000
0.	0.	.3000	.6000	.6000

VARIABLE ---- WFTAAW(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF TORPEDOS THAT ARE TARGETED AGAINST AAW SHIPS.

1.000

VARIABLE ---- WFTASW(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF TORPEDOS THAT ARE TARGETED AGAINST ASW SHIPS.

1.000

VARIABLE ---- WFTFL (5, 0, 0)
(L) WEIGHTING FACTOR APPLIED TO CARRIER EFFECTIVENESS (XEFFCM) WHEN THE TASK FORCE IS IN LOCATION L TO DETERMINE THE CUMULATIVE WEIGHTED NUMBER OF EFFECTIVE TIME PERIODS THAT THE CARRIERS SPENT ON STATION. IF THERE ARE NO CARRIERS (XPLAT=0), THEN THIS FACTOR IS APPLIED TO THE TOTAL NUMBER OF BLUE SHIPS REMAINING IN THE TASK FORCE.

0.	0.	1.000	1.000	1.000

VARIABLE ---- WFTPLT(1, 0, 0)
WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF TORPEDOS THAT ARE TARGETED AGAINST AIRCRAFT CARRIERS.

1.000

```
VARIABLE ---- WFTURG(  1,  0,  0)
                                   WEIGHTING FACTOR USED IN DETERMINING THE PROPORTION OF TORPEDOS THAT
                                   ARE TARGETED AGAINST URG SHIPS.
         1.000

VARIABLE ---- WRLNDQ(  5,  0,  0)
                                   (L) RANGE (OF THE AIR ATTACK FROM TASK-FORCE CENTER) THAT LAND-BASED
                                   RESOURCES ARE ABLE TO WARN THE TASK FORCE (IN LOCATION L) OF AN
                                   IMPENDING RED AIR ATTACK GIVEN THAT THESE LAND RESOURCES ARE ABLE TO
                                   PROVIDE THIS WARNING. (NMI)
         800.0       700.0           500.0       300.0       200.0

VARIABLE ---- WTFCBO(  1,  0,  0)
                                   WIDTH TASK FORCE USES IN CROSSING BARRIER, IF BARRIER CROSSING PROTO-
                                   COL 3 IS USED. (NMI)
         50.00

VARIABLE ---- WVSIZ (  1,  0,  0)
                                   THE NUMBER OF FIGHTERS SIMULTANEOUSLY LAUNCHED IN EACH DLI WAVE.
         2.000

VARIABLE ---- XAEWLQ(  5,  0,  0)
                                   (L) THE NUMBER OF LAND-BASED AEW AIRCRAFT ASSIGNED TO THE TASK FORCE
                                   WHEN THE TASK FORCE IS IN LOCATION L.
         30.00       30.00           30.00       15.00       0.

VARIABLE ---- XAEW  (  1,  0,  0)
                                   THE NUMBER OF SEA-BASED AEW AIRCRAFT ASSIGNED TO THE TASK FORCE.
         20.00

VARIABLE ---- XASWLQ(  5,  0,  0)
                                   (L) THE NUMBER OF LAND-BASED ASW AIRCRAFT ASSIGNED TO THE TASK FORCE
                                   WHEN THE TASK FORCE IS IN LOCATION L.
         30.00       30.00           30.00       15.00       0.

VARIABLE ---- XASW  (  1,  0,  0)
                                   THE NUMBER OF SEA-BASED ASW AIRCRAFT ASSIGNED TO THE TASK FORCE.
         20.00

VARIABLE ---- XATTCK(  1,  0,  0)
                                   NUMBER OF ATTACK  AIRCRAFT BASED ON ALL OF THE AIRCRAFT CARRIERS IN
                                   THE TASK FORCE.
         72.00

VARIABLE ---- XEAAW (  1,  0,  0)
                                   THE NUMBER OF AAW ESCORTS.
         8.000
```

```
VARIABLE ---- XEASWA(  1,   0,   0)
                                          THE NUMBER OF AIR-CAPABLE ASW ESCORTS.
          6.000

VARIABLE ---- XEASWN(  1,   0,   0)
                                          THE NUMBER OF NON-AIR-CAPABLE ASW ESCORTS.
          2.000

VARIABLE ---- XFGHTR(  1,   0,   0)
                                          NUMBER OF FIGHTER AIRCRAFT BASED ON ALL OF THE AIRCRAFT CARRIERS IN
                                          THE TASK FORCE.
          48.00

VARIABLE ---- XIA   (  5,   0,   0)
                                          (L) MINIMUM NUMBER OF BLUE SORTIES  ON ATTACK MISSION NEEDED TO PER-
                                          FORM AIRBASE ATTACK WHEN THE TASK FORCE IS IN LOCATION L.
          10.00        100.0               10.00        10.00        0.

VARIABLE ---- XIE   (  5,   0,   0)
                                          (L) MINIMUM NUMBER OF BLUE SORTIES  ON ESCORT MISSION NEEDED TO PER-
                                          FORM AIRBASE ATTACK WHEN THE TASK FORCE IS IN LOCATION L.
          100.0        10.00               0.           0.           0.

VARIABLE ---- XNRAB (  1,   0,   0)
                                          NUMBER OF ACTUAL AIRBASES THAT COMRISE THE NOTIONAL VULNERABLE RED
                                          AIRBASE.
          10.00

VARIABLE ---- XPLAT (  1,   0,   0)
                                          THE NUMBER OF CARRIERS IN THE TASK FORCE.
          2.000

VARIABLE ---- XURGS (  1,   0,   0)
                                          NUMBER OF URG SHIPS IN THE TASK FORCE.
          4.000

VARIABLE ---- ZLAMPF(  1,   0,   0)
                                          THE AVERAGE NUMBER OF REACTIVE VTOL ASW AIRCRAFT SORTIES PER AIR-
                                          CAPABLE ASW ESCORT AVAILABLE FOR THE PROSECUTION OF THE TORPEDO
                                          SUBMARINE THREAT.
          2.000

VARIABLE ---- ZMPATT(  3,   0,   0)
                                          (K) THE NUMBER OF ASMS CARRIED BY RED BOMBERS OF TYPE K.
          2.000        3.000               1.000

VARIABLE ---- ZMPCAP(  1,   0,   0)
                                          THE NUMBER OF MISSILE SALVQS THAT WILL BE FIRED BY ENGAGING CAP
                                          AIRCRAFT.
          3.000
```

VARIABLE ---- ZMPDLI(1, 0, 0)
THE NUMBER OF MISSILE SALVOS THAT WILL BE FIRED BY ENGAGING DLI.

3.000

VARIABLE ---- ZMPESC(1, 0, 0)
NUMBER OF MISSLE SALVOS THAT WILL BE FIRED BY ENGAGING RED ESCORT AIRCRAFT.

2.000

VARIABLE ---- ZMPSTG(1, 0, 0)
THE NUMBER OF MISSILES FIRED BY EACH SURVIVING MISSILE SUBMARINE.

5.000

APPENDIX D

SAMPLE OUTPUT OF MEDMOD RESULTS

SAMPLE OUTPUT OF MEDMOD RESULTS

MEDMOD currently provides three types of outputs, and all three are produced in each run of MEDMOD. First, MEDMOD, through Overlay INP, provides an output listing of the inputs. (A sample of this output is given in Appendix C.) Next, MEDMOD provides some detailed outputs which essentially consist of the results of each major subroutine (and of some of the subroutines called by these major subroutines) each time these subroutines are called. Finally, MEDMOD provides a short summary results table.

The following is a sample of the detailed output and the summary results table produced by MEDMOD given the data listed in Appendix C. The detailed output appears first, and the summary results table is on the last page of this appendix. (For brevity, the detailed output of time periods three through eight has been deleted here.)

```
OUTPUTS FROM THIS RUN OF MEDMOD ARE AS FOLLOWS

START PROGRAM MEDMOD

--------------------------------------------------
START SUBROUTINE DDAY
RESULTS OF THE DDAY SHOOTOUT--TASK FORCE IS IN REGION  1

KIND OF RED SHIP                                   1        2        3        4
INITIAL RED SHIPS IN REGION                    0.000    3.000    0.000    2.000
SHIPS KILLED BEFORE ATTACK ON TASK FORCE       0.000    1.000    0.000    1.000
SHIPS ATTACKING CARRIERS                       0.000     .600    0.000     .400
SHIPS ATTACKING OTHER BLUE SHIPS               0.000     .400    0.000     .600
SHIPS KILLED AFTER ATTACK ON TASK FORCE        0.000    1.000    0.000    1.000
RESULTANT RED SHIPS IN REGION                  0.000    1.000    0.000    0.000

KIND OF BLUE SHIP                              XPLAT    XEAAW   XEASWA   XEASWN    XURGS
INITIAL BLUE SHIPS IN REGION                   2.000    8.000    6.000    2.000    4.000
BLUE SHIPS KILLED                              0.000     .058     .044     .015     .029
RESULTANT BLUE SHIPS IN REGION                 2.000    7.942    5.956    1.985    3.971
RESULTANT RELATIVE CARRIER CAPABILITY (XEFFCM) EQUALS   .9560.
 ENACD=     .1250  PIACD=      .0010   FACD=      .0017
 FDMCV=     .0600 XFGHTR=   47.9200
 ADMCV=     .1200 XATTCK=   71.8800
END OF SUBROUTINE DDAY
--------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------

START OF PERIOD     1

 LOCTF= 1

--------------------------------------------------
START SUBROUTINE GNAATK
INSUFFICIENT RED BOMBERS. NO AIR ATTACK ON TASK FORCE.
   BMT=    0.0000     EST=    0.0000 NTPSLA=    1 ITP=    1

THE FOLLOWING VALUES ARE FOR I = 1
  ESC(I)=     0.0000
BMR(I,1)=     0.0000
BMR(I,2)=     0.0000
BMR(I,3)=     0.0000
AIRCRAFT ON GROUND ON AIRBASE I = 1 --
 AESCAB(I)=  100.0000
ATABT(I,1)=   40.0000
ATABT(I,2)=   20.0000
ATABT(I,3)=   20.0000

THE FOLLOWING VALUES ARE FOR I = 2
  ESC(I)=     0.0000
BMR(I,1)=     0.0000
BMR(I,2)=     0.0000
BMR(I,3)=     0.0000
AIRCRAFT ON GROUND ON AIRBASE I = 2 --
 AESCAB(I)=  100.0000
ATABT(I,1)=   40.0000
ATABT(I,2)=   20.0000
ATABT(I,3)=   20.0000
END OF SUBROUTINE GNAATK
--------------------------------------------------
```

```
START SUBROUTINE PLBAB
NO RED AIR ATTACK ON TASK FORCE THIS PERIOD
END OF SUBROUTINE PLBAB
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE SUBSUB
TASK FORCE IS IN REGION  1
RESULTS OF THE BLUE SUBMARINE/RED SUBMARINE INTERACTION
    INITIAL BLUE SUBMARINES IN TASK FORCE-----   4.00          INITIAL RED SUBMARINES IN REGION----------   1.00
     (ALL BLUE SUBS ENGAGE IN COMBAT.)                          RED SUBS ENGAGING IN COMBAT---------------    .90
    RED SUBMARINES CAPABLE OF ATTACKING BLUE--    .82          BLUE SUBMARINES CAPABLE OF ATTACKING RED--   3.48
    BLUE SUBMARINES KILLED BY RED SUBMARINES--    .11          RED SUBMARINES KILLED BY BLUE SUBMARINES--    .67
    RESULTANT BLUE SUBMARINES IN TASK FORCE---   3.89          RESULTANT RED SUBMARINES IN REGION--------    .33
NO RED SURFACE SHIPS IN REGION--BLUE SUB/RED SURFACE SHIP BATTLE DOES NOT TAKE PLACE.
OVERALL BLUE RESULTS--   4.00 BLUE SSN(DS) INITIALLY LESS    .11 KILLED YIELDS   3.89 SURVIVING.
OVERALL RED RESULTS, BY KIND OF RED SHIP.  (ATTRITION IS PROPORTIONAL.)
    KIND OF RED SHIP                  1      2      3      4
    INITIAL RED SHIPS IN REGION     0.00   1.00   0.00   0.00
    RED SHIPS KILLED                0.00    .67   0.00   0.00
    RESULTANT RED SHIPS IN REGION   0.00    .33   0.00   0.00
END OF SUBROUTINE SUBSUB
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE CTFMOD
ATT=    0.0000

 UBAEWL    UBAEW
   .1400    .1400

 AEWD      STAR
  220.00    300.00

 CAPML     CAPM      BUCAP     DLIA      WVSIZ
    0.00     1.00      6.00       .67      2.00

 T1        T2        T3        T4
    8.00     3.00      0.00      1.00

 VCAP      CAPMR     TCAP      CAPSTAR
   12.00    50.00      1.50    400.00

 BARL
  500.00

 UBASWL    BAREAL
   .1000 1900.0000

 UBASW     BAREA
   .2500 1900.0000

 ASWF      PKASW     ST        PDIN      PKIN      ZLAMPF
   .0500    .5000    0.0000     .2000     .5000    2.0000

 PKIIN     ESR       ESLR      SUBSOR
     .15     8.00      8.00     15.00

 TPS
    2.50

 ZMPCAP    ZMPDLI
    3.00     3.00

 STG       ZMPSTC
     .03     5.00
```

```
STSALV
    2.00

XASW       XAEW       XASWL      XAEWL
   20.00      20.00      30.00      30.00

XEASWA     XEASWN     XEAAW
    5.96       1.99       7.94

XPLAT
    2.00
  XD             29.56
  XPDASW           .77
  XSURS1          0.00
  XSTG1            .02
  XSURS2          0.00
  XTOTE          31.77
  XSURS3          0.00
  PS(1,L)=        0.0000 RS(2,L)=         .3179 FOR L = 1
  XSALVS          0.00
  XPST            1.00
  XAEWSTA         7.00
  XCAPSTA         2.80
  XZ            516.03
  XZ            800.00

START OF ITERATION I= 1 THROUGH ATTRITION PORTION OF CTFMOD
ATT=     0.0000

DISPLAY RESULTS OF ASM-VS-SHIP BATTLE
  XATS1           .10
FMRBT(1)=    0.0000 FMRBT(2)=    0.0000 FMRBT(3)=    0.0000 FMRBT(
FMPBT(NKRB+1)=    1.0000

PKSS       FPPL1      PKPL1      PKPL2      PKPLD      FPPL2
     .75       7.00        .75        .33        .50       1.00

TAB10
  8.0000   16.0000   24.0000   32.0000   40.0000   48.0000   56.0000   64.0000
 72.0000   80.0000   88.0000   96.0000  104.0000  112.0000  120.0000  128.0000
136.0000  144.0000  152.0000  160.0000

TAB13
   .6000     .3000     .1500     .0750     .0350     .0150     .0100     .0050
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
  0.0000    0.0000    0.0000    0.0000
  XATS2           .03
  XMPPLAT         .01
  XATS3           .00
  XATS4           .00
ENACD=     0.0000  PIACD=     0.0000    FACD=     0.0000
FDMCV=     0.0000 XFGHTR=    47.9200
ADMCV=     0.0000 XATTCK=    71.8800
  XPSA           1.00
  XEFFCM          .96
 PMAAW      .0001    PMASW      .0006    PMURG      .0006    PTAAW     0.0000    PTASW     0.0000    PTURG     0.0000
SURAAW      .9999   SURASW      .9994   SURURG      .9994
 XEAAW     7.9411   XEASWA     5.9528   XEASWN     1.9843    XURGS     3.9683

START OF ITERATION I= 2 THROUGH ATTRITION PORTION OF CTFMOD
ATT=     0.0000

DISPLAY RESULTS OF AIR-TO-AIR  BATTLE
  AESC=     0.0000 XFGHTR=    47.9200
 AT(1)=     0.0000
```

```
 AT(2)=    0.0000
 AT(3)=    0.0000
 FTSORU=    0.0000  ATSORU=    0.0000
ATABT(1,KRB)=   40.0000 ATABT(2,KRB)=   40.0000 FOR KRB = 1
ATABT(1,KRB)=   20.0000 ATABT(2,KRB)=   20.0000 FOR KRB = 2
ATABT(1,KRB)=   20.0000 ATABT(2,KRB)=   20.0000 FOR KRB = 3
   AESCAB(1)=  100.0000    AESCAB(2)=  100.0000
RELATIVE CARRIER CAPABILITY (XEFFCM) IS NOW  .9557.
END OF SUBROUTINE CTFMOD
--------------------------------------------------

REMINDER--THIS IS PERIOD   1, TASK FORCE IS IN REGION  1.

--------------------------------------------------
START SUBROUTINE SHPSHP
NO VULNERABLE RED SURFACE SHIPS IN REGION, HENCE NO COMBAT TAKES PLACE.
    KIND OF BLUE SHIP                      XPLAT  XEAAW XEASWA XEASWN  XURGS
    BLUE SHIPS IN TASK FORCE                2.00   7.94   5.95   1.98   3.97
    RELATIVE CARRIER CAPABILITY (XEFFCM) EQUALS  .9557.
END OF SUBROUTINE SHPSHP
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE POWERP
TASK FORCE IS IN REGION  1.
POWER PROJECTION SORTIE RATES FOR THIS REGION ARE ZERO.  NO POWER PROJECTION PERFORMED.
END OF SUBROUTINE POWERP
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE ADDMOE
 XPLAT=    2.0000 XEFFCM=     .9557  XSHIP=   21.8465  ISTOP= 0    ITP=   1
END OF SUBROUTINE ADDMOE
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE MOVTF
TASK FORCE DOES NOT MOVE DURING PERIOD   1.  IT REMAINS IN REGION  1.
END OF SUBROUTINE MOVTF
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE MOVRS
ATTRITION TO RED SHIPS TRANSITING BLUE-CONTROLLED BARRIERS DURING PERIOD    1, BY KIND OF RED SHIP
    BARRIER BETWEEN REGIONS  1 AND  2
       RED SHIPS ATTEMPTING TRANSIT           .29   0.00   0.00   6.08
       RED SHIPS KILLED                       .03   0.00   0.00   2.11
       BLUE BARRIER SUBMARINES--  10.00 INITIALLY LESS   5.53 COUNTERKILLED YIELDS   4.47 SURVIVING.
    BARRIER BETWEEN REGIONS  2 AND  3
       RED SHIPS ATTEMPTING TRANSIT           .90    .65   0.00   6.49
       RED SHIPS KILLED                       .07    .05   0.00   1.66
       BLUE BARRIER SUBMARINES--  10.00 INITIALLY LESS   8.46 COUNTERKILLED YIELDS   1.54 SURVIVING.
FLOW OF RED SHIPS DURING PERIOD     1, BY KIND OF RED SHIP
    REGION 1
       INITIAL RED SHIPS IN REGION           0.00    .32   0.00   0.00
       RED SHIPS ENTERING REGION              .26   0.00   0.00   3.97
       RED SHIPS LEAVING REGION              0.00   0.00   0.00   0.00
       RESULTANT RED SHIPS IN REGION          .26    .32   0.00   3.97
    REGION 2
       INITIAL RED SHIPS IN REGION            .50   0.00   0.00   8.00
       RED SHIPS ENTERING REGION              .83    .60   0.00   4.83
       RED SHIPS LEAVING REGION               .29   0.00   0.00   6.08
       RESULTANT RED SHIPS IN REGION         1.04    .60   0.00   6.75
    REGION 3
       INITIAL RED SHIPS IN REGION           1.00    .70   0.00  11.00
       RED SHIPS ENTERING REGION              .19    .07    .66    .22
```

```
      RED SHIPS LEAVING REGION                 .90     .65    0.00    6.49
      RESULTANT RED SHIPS IN REGION            .29     .12     .66    4.73
   REGION 4
      INITIAL RED SHIPS IN REGION             2.20    1.20   15.00    5.00
      RED SHIPS ENTERING REGION                .15     .06    1.13     .98
      RED SHIPS LEAVING REGION                 .19     .07     .66     .22
      RESULTANT RED SHIPS IN REGION           2.15    1.19   15.47    5.76
   REGION 5
      INITIAL RED SHIPS IN REGION             3.00    3.00   25.00   20.00
      RED SHIPS ENTERING REGION               0.00    0.00    0.00    0.00
      RED SHIPS LEAVING REGION                 .15     .06    1.13     .98
      RESULTANT RED SHIPS IN REGION           2.85    2.94   23.88   19.02
END OF SUBROUTINE MOVRS
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE ABATCK
TASK FORCE IS NOW IN REGION  1
INSUFFICIENT BLUE ESCORT AIRCRAFT--NO AIRBASE ATTACK.
END OF SUBROUTINE ABATCK
--------------------------------------------------

END OF PERIOD    1

----------------------------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------------------------

START OF PERIOD    2

 LOCTF= 1

--------------------------------------------------
START SUBROUTINE GNAATK
NO RED AIR ATTACK ON TASK FORCE SCHEDULED THIS PERIOD.
   BMT=    0.0000    EST=    0.0000 NTPSLA=    2 ITP=    2

THE FOLLOWING VALUES ARE FOR I = 1
  ESC(I)=     0.0000
BMR(I,1)=     0.0000
BMR(I,2)=     0.0000
BMR(I,3)=     0.0000
AIRCRAFT ON GROUND ON AIRBASE I = 1 --
 AESCAB(I)=  100.0000
ATABT(I,1)=   40.0000
ATABT(I,2)=   20.0000
ATABT(I,3)=   20.0000

THE FOLLOWING VALUES ARE FOR I = 2
  ESC(I)=     0.0000
BMR(I,1)=     0.0000
BMR(I,2)=     0.0000
BMR(I,3)=     0.0000
AIRCRAFT ON GROUND ON AIRBASE I = 2 --
 AESCAB(I)=  100.0000
ATABT(I,1)=   40.0000
ATABT(I,2)=   20.0000
ATABT(I,3)=   20.0000
END OF SUBROUTINE GNAATK
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE PLBAB
NO RED AIR ATTACK ON TASK FORCE THIS PERIOD
END OF SUBROUTINE PLBAB
--------------------------------------------------
```

```
--------------------------------------------------
START SUBROUTINE SUBSUB
TASK FORCE IS IN REGION  1
RESULTS OF THE BLUE SUBMARINE/RED SUBMARINE INTERACTION
    INITIAL BLUE SUBMARINES IN TASK FORCE-----   3.89        INITIAL RED SUBMARINES IN REGION----------    .58
     (ALL BLUE SUBS ENGAGE IN COMBAT.)                       RED SUBS ENGAGING IN COMBAT---------------    .52
    RED SUBMARINES CAPABLE OF ATTACKING BLUE--    .47        BLUE SUBMARINES CAPABLE OF ATTACKING RED--   3.39
    BLUE SUBMARINES KILLED BY RED SUBMARINES--    .08        RED SUBMARINES KILLED BY BLUE SUBMARINES--    .31
    RESULTANT BLUE SUBMARINES IN TASK FORCE---   3.81        RESULTANT RED SUBMARINES IN REGION--------    .27
RESULTS OF THE BLUE SUBMARINE/RED SURFACE SHIP INTERACTION
    BLUE SUBS ATTACKING RED SURFACE SHIPS-----   3.81        INITIAL RED SURFACE SHIPS IN REGION-------   3.97
     (ALL BLUE SUBS THAT SURVIVED RED SUBS)                  RED SURFACE SHIPS ENGAGING IN COMBAT------   3.38
    RED SURF.SHIPS CAPABLE OF ATTACKING BLUE--   2.06        BLUE SUBS CAPABLE OF ATTACKING RED SURF.--   3.28
    BLUE SUBS KILLED BY RED SURFACE SHIPS-----    .46        RED SURFACE SHIPS KILLED------------------   1.06
    RESULTANT BLUE SUBMARINES IN TASK FORCE---   3.36        RESULTANT RED SURFACE SHIPS IN REGION-----   2.92
OVERALL BLUE RESULTS--   3.89 BLUE SSN(DS) INITIALLY LESS    .53 KILLED YIELDS   3.36 SURVIVING.
OVERALL RED RESULTS, BY KIND OF RED SHIP.  (ATTRITION IS PROPORTIONAL.)
    KIND OF RED SHIP                    1      2      3      4
    INITIAL RED SHIPS IN REGION       .26    .32   0.00   3.97
    RED SHIPS KILLED                  .14    .17   0.00   1.06
    RESULTANT RED SHIPS IN REGION     .12    .15   0.00   2.92
END OF SUBROUTINE SUBSUB
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE CTFMOD
ATT=     0.0000

 UBAEWL    UBAEW
   .1400     .1400

 AEWD      STAR
  220.00    300.00

 CAPML     CAPM      BUCAP     DLIA      WVSIZ
    0.00      1.00      6.00       .67      2.00

 T1        T2        T3        T4
    8.00      3.00      0.00      1.00

 VCAP      CAPMR     TCAP      CAPSTAR
   12.00     50.00      1.50    400.00

 BARL
  500.00

 UBASWL    BAREAL
   .1000 1900.0000

 UBASW     BAREA
   .2500 1900.0000

 ASWF      PKASW     ST        PDIN       PKIN      ZLAMPF
   .0500     .5000     .0122     .2000     .5000    2.0000

 PKIIN     ESR       FSLR      SUBSOR
     .15      8.00      8.00     15.00

 TPS
    2.50

 ZMPCAP    ZMPDLI
    3.00      3.00

 SIG       ZMPSIG
```

```
      .01       5.00

 STSALV
     2.00

 XASW      XAEW      XASWL     XAEWL
    20.00     20.00     30.00     30.00

 XEASWA    XEASWN    XEAAW
     5.95      1.98      7.94

 XPLAT
     2.00
   XO               29.56
   XPDASW             .77
   XSURS1             .01
   XSTG1              .01
   XSURS2             .01
   XTOTE            31.75
   XSURS3             .01
   RS(1,L)=           .1155 RS(2,L)=          .1434 FOR L = 1
   XSALVS             .01
   XPST              1.00
   XAEWSTA           7.00
   XCAPSTA           2.80
   XZ              516.03
   XZ              800.00

START OF ITERATION I= 1 THROUGH ATTRITION PORTION OF CTFMOD
ATT=    0.0000

DISPLAY RESULTS OF ASM-VS-SHIP BATTLE
   XATS1              .05
FMRBT(1)=    0.0000 FMRBT(2)=    0.0000 FMRBT(3)=    0.0000 FMRBT(
FMRBT(NKRB+1)=    1.0000

 PKSS      FPPL1     PKPL1     PKPL2     PKPLD     FPPL2
      .75      7.00       .75       .33       .50      1.00

 TAB10
   8.0000   16.0000   24.0000   32.0000   40.0000   48.0000   56.0000   64.0000
  72.0000   80.0000   88.0000   96.0000  104.0000  112.0000  120.0000  128.0000
 136.0000  144.0000  152.0000  160.0000

 TAB13
    .6000     .3000     .1500     .0750     .0350     .0150     .0100     .0050
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000
   XATS2              .01
   XMPPLAT            .00
   XATS3              .00
   XATS4              .00
 ENACD=    0.0000  PIACD=    0.0000   FACD=    0.0000
 FDMCV=    0.0000 XFGHTR=   47.9200
 ADMCV=    0.0000 XATTCK=   71.8800
   XPSA              1.00
   XEFFCM             .96
  PMAAW      .0000    PMASW      .0003   PMURG      .0003   PTAAW      .0013   PTASW      .0013   PTURG      .0013
 SURAAW      .9967  SURASW      .9984  SURURG      .9984
  XEAAW     7.9305  XEASWA     5.9435  XEASWN     1.9812   XURGS     3.9620

START OF ITERATION I= 2 THROUGH ATTRITION PORTION OF CTFMOD
ATT=    0.0000

DISPLAY RESULTS OF AIR-TO-AIR  BATTLE
  AFSC=    0.0000 XFGHTR=   47.9200
```

```
 AT(1)=    0.0000
 AT(2)=    0.0000
 AT(3)=    0.0000
 FTSORU=    0.0000  ATSORU=    0.0000
ATABT(1,KRB)=   40.0000 ATABT(2,KRB)=   40.0000 FOR KRB = 1
ATABT(1,KRB)=   20.0000 ATABT(2,KRB)=   20.0000 FOR KRB = 2
ATABT(1,KRB)=   20.0000 ATABT(2,KRB)=   20.0000 FOR KRB = 3
   AESCAB(1)=  100.0000    AESCAB(2)=  100.0000
RELATIVE CARRIER CAPABILITY (XEFFCM) IS NOW  .9553.
END OF SUBROUTINE CTFMOD
--------------------------------------------------

REMINDER--THIS IS PERIOD   2, TASK FORCE IS IN REGION  1.

--------------------------------------------------
START SUBROUTINE SHPSHP
BLUE AIRCRAFT FROM CARRIER--   92.02.  BLUE AIRCRAFT REQUIRED TO DESTROY ALL VULNERABLE RED SHIPS--   29.16.
 XATTCK=   70.4219,  AAK=    1.4582,  ATSORU=   27.7053
 XFGHTR=   47.9200,  FAK=    0.0000,  FTSORU=    0.0000
SINCE THERE ARE SUFFICIENT BLUE AIRCRAFT TO DESTROY ALL VULNERABLE RED SHIPS, THERE IS NO ATTRITION TO BLUE SHIPS.
(THE SHIP-TO-SHIP INTERACTION DOES NOT TAKE PLACE.)
    KIND OF BLUE SHIP                     XPLAT  XEAAW XEASWA XEASWN  XURGS
    BLUE SHIPS IN TASK FORCE               2.00   7.93   5.94   1.98   3.96
    RELATIVE CARRIER CAPABILITY (XEFFCM) EQUALS  .9553.
RED SURFACE SHIP RESULTS (NOTE--RED SHIP KINDS 1 AND 2 ARE SUBMARINES, WHICH DO NOT PARTICIPATE IN THIS INTERACTION.)
    KIND OF RED SHIP                          3      4
    INITIAL RED SHIPS IN REGION            0.00   2.92
    RED SHIPS VULNERABLE TO BLUE ATTACK    0.00   2.92
    RED SHIPS DESTROYED BY BLUE AIRCRAFT   0.00   2.92
    RESULTANT RED SHIPS IN REGION          0.00   0.00
END OF SUBROUTINE SHPSHP
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE POWERP
TASK FORCE IS IN REGION  1.
POWER PROJECTION SORTIE RATES FOR THIS REGION ARE ZERO.  NO POWER PROJECTION PERFORMED.
END OF SUBROUTINE POWERP
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE ADDMOE
 XPLAT=    2.0000 XEFFCM=     .9553  XSHIP=   21.8171  ISTOP= 0    ITP=   2
END OF SUBROUTINE ADDMOE
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE MOVTF
DURING PERIOD   2 TASK FORCE MOVES FROM REGION  1 TO REGION  2.
BARRIER BETWEEN REGIONS  1 AND  2 IS CONTROLLED BY BLUE, HENCE THERE IS NO ATTRITION TO THE TASK FORCE.
END OF SUBROUTINE MOVTF
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE MOVRS
ATTRITION TO RED SHIPS TRANSITING BLUE-CONTROLLED BARRIERS DURING PERIOD    2, BY KIND OF RED SHIP
    BARRIER BETWEEN REGIONS  1 AND  2
       RED SHIPS ATTEMPTING TRANSIT          .04    .07   0.00   0.00
       RED SHIPS KILLED                      .00    .01   0.00   0.00
       BLUE BARRIER SUBMARINES--   4.47 INITIALLY LESS    .01 COUNTERKILLED YIELDS   4.46 SURVIVING.
    BARRIER BETWEEN REGIONS  2 AND  3
       RED SHIPS ATTEMPTING TRANSIT          .09    .05    .42   1.80
       RED SHIPS KILLED                      .00    .00    .04    .20
       BLUE BARRIER SUBMARINES--   1.54 INITIALLY LESS    .81 COUNTERKILLED YIELDS    .74 SURVIVING.
FLOW OF RED SHIPS DURING PERIOD    2, BY KIND OF RED SHIP
    REGION 1
```

```
        INITIAL RED SHIPS IN REGION          .12     .14    0.00    0.00
        RED SHIPS ENTERING REGION           0.00    0.00    0.00    0.00
        RED SHIPS LEAVING REGION             .04     .07    0.00    0.00
        RESULTANT RED SHIPS IN REGION        .07     .07    0.00    0.00
     REGION 2
        INITIAL RED SHIPS IN REGION         1.04     .60    0.00    6.75
        RED SHIPS ENTERING REGION            .13     .11     .39    1.59
        RED SHIPS LEAVING REGION            0.00    0.00    0.00    0.00
        RESULTANT RED SHIPS IN REGION       1.17     .71     .39    8.34
     REGION 3
        INITIAL RED SHIPS IN REGION          .29     .12     .66    4.73
        RED SHIPS ENTERING REGION            .14     .01     .02     .35
        RED SHIPS LEAVING REGION             .09     .05     .42    1.80
        RESULTANT RED SHIPS IN REGION        .34     .09     .25    3.28
     REGION 4
        INITIAL RED SHIPS IN REGION         2.15    1.19   15.47    5.76
        RED SHIPS ENTERING REGION            .19     .25     .17    1.69
        RED SHIPS LEAVING REGION             .14     .01     .02     .35
        RESULTANT RED SHIPS IN REGION       2.20    1.43   15.62    7.11
     REGION 5
        INITIAL RED SHIPS IN REGION         2.85    2.94   23.88   19.02
        RED SHIPS ENTERING REGION           0.00    0.00    0.00    0.00
        RED SHIPS LEAVING REGION             .19     .25     .17    1.69
        RESULTANT RED SHIPS IN REGION       2.67    2.69   23.71   17.33
END OF SUBROUTINE MOVRS
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE ABATCK
TASK FORCE IS NOW IN REGION  2
INSUFFICIENT BLUE ATTACK AIRCRAFT--NO AIRBASE ATTACK.
END OF SUBROUTINE ABATCK
--------------------------------------------------

END OF PERIOD    2

------------------------------------------------------------------------------------------------------------------------
```

FOR BREVITY, THE RESULTS OF PERIODS 3 THROUGH 8 HAVE BEEN DELETED.

```
------------------------------------------------------------------------------------------------------------------------

START OF PERIOD    9

 LOCTF= 5

--------------------------------------------------
START SUBROUTINE GNAATK
   BMT=   11.4801    FST=   54.2202 NTPSLA=    1 ITP=    9

THE FOLLOWING VALUES ARE FOR I = 1
  ESC(I)=     6.0982
BMR(I,1)=     1.7999
BMR(I,2)=      .8193
BMR(I,3)=      .2572
AIRCRAFT ON GROUND ON AIRBASE I = 1 --
 AESCAB(I)=     1.5245
ATABT(I,1)=      .4500
ATABT(I,2)=      .2048
ATABT(I,3)=      .0286

THE FOLLOWING VALUES ARE FOR I = 2
  FSC(I)=    48.1220
BMR(I,1)=     4.8014
```

```
BMR(I,2)=    1.7146
BMR(I,3)=    2.0877
AIRCRAFT ON GROUND ON AIRBASE I = 2 --
 AESCAB(I)=   12.0305
ATABT(I,1)=    1.2003
ATABT(I,2)=     .4286
ATABT(I,3)=     .5219
END OF SUBROUTINE GNAATK
-------------------------------------------------

-------------------------------------------------
START SUBROUTINE PLBAB
------------------------
START SUBROUTINE AIRAIR
   E(1)=   54.2202
  EA(1)=    4.9069   EK(1)=    32.2390   EH(1)=   17.0742
   D(1)=   77.7028
 DA1(1)=   58.2771  DK1(1)=     4.4570  DH1(1)=   14.9687
 DA2(1)=    0.0000  DK2(1)=     0.0000  DH2(1)=   58.2771
  DA(1)=    0.0000   DK(1)=     4.4570   DH(1)=   73.2457
   D(2)=  119.5502
 DA1(2)=    0.0000  DK1(2)=    13.2384  DH1(2)=  106.3118
 DA2(2)=    0.0000  DK2(2)=     0.0000  DH2(2)=    0.0000
  DA(2)=    0.0000   DK(2)=    13.2384   DH(2)=  106.3118
   A(1)=    6.6013
  AA(1)=     .0000   AK(1)=     6.6013   AH(1)=    0.0000
   A(2)=    2.5339
  AA(2)=     .0000   AK(2)=     2.5339   AH(2)=    0.0000
   A(3)=    2.3449
  AA(3)=     .0001   AK(3)=     2.3448   AH(3)=    0.0000
END OF SUBROUTINE AIRAIR
------------------------
  BMR(1,KRB)=      .0000   BMR(2,KRB)=      .0000 FOR KRB = 1
  BMR(1,KRB)=      .0000   BMR(2,KRB)=      .0000 FOR KRB = 2
  BMR(1,KRB)=      .0000   BMR(2,KRB)=      .0001 FOR KRB = 3
      ESC(1)=      .5519       ESC(2)=     4.3550
ATABT(1,KRB)=      .4500 ATABT(2,KRB)=     1.2003 FOR KRB = 1
ATABT(1,KRB)=      .2048 ATABT(2,KRB)=      .4286 FOR KRB = 2
ATABT(1,KRB)=      .0286 ATABT(2,KRB)=      .5219 FOR KRB = 3
   AESCAB(1)=     3.4449    AESCAB(2)=    27.1844
 PLBLBD(KBD,LB) --
 PLBLBD(1,1)=    0.0000
 PLBLBD(1,2)=   61.2298
 PLBLBD(1,3)=    0.0000
 PLBLBD(1,4)=    0.0000
 PLBLBD(1,5)=   18.4722
 PLBLBD(2,1)=    0.0000
 PLBLBD(2,2)=   43.5649
 PLBLBD(2,3)=    0.0000
 PLBLBD(2,4)=   26.5021
 PLBLBD(2,5)=   37.8448
END OF SUBROUTINE PLBAB
-------------------------------------------------

-------------------------------------------------
START SUBROUTINE SUBSUB
TASK FORCE IS IN REGION  5
RESULTS OF THE BLUE SUBMARINE/RED SUBMARINE INTERACTION
    INITIAL BLUE SUBMARINES IN TASK FORCE-----      .03
     (ALL BLUE SUBS ENGAGE IN COMBAT.)
    RED SUBMARINES CAPABLE OF ATTACKING BLUE--    3.46
    BLUE SUBMARINES KILLED BY RED SUBMARINES--     .00
    RESULTANT BLUE SUBMARINES IN TASK FORCE---     .03
RESULTS OF THE BLUE SUBMARINE/RED SURFACE SHIP INTERACTION
    BLUE SUBS ATTACKING RED SURFACE SHIPS-----      .03
     (ALL BLUE SUBS THAT SURVIVED RED SUBS)

INITIAL RED SUBMARINES IN REGION----------    4.05
RED SUBS ENGAGING IN COMBAT---------------    3.80
BLUE SUBMARINES CAPABLE OF ATTACKING RED--     .02
RED SUBMARINES KILLED BY BLUE SUBMARINES--     .01
RESULTANT RED SUBMARINES IN REGION--------    4.04

INITIAL RED SURFACE SHIPS IN REGION-------   38.50
RED SURFACE SHIPS ENGAGING IN COMBAT------   34.26
```

```
    RED SURF.SHIPS CAPABLE OF ATTACKING BLUE--  30.49         BLUE SUBS CAPABLE OF ATTACKING RED SURF.--     .02
    BLUE SUBS KILLED BY RED SURFACE SHIPS-----    .01         RED SURFACE SHIPS KILLED-----------------      .01
    RESULTANT BLUE SUBMARINES IN TASK FORCE---    .02         RESULTANT RED SURFACE SHIPS IN REGION-----   38.49
OVERALL BLUE RESULTS--     .03 BLUE SSN(DS) INITIALLY LESS     .01 KILLED YIELDS     .02 SURVIVING.
OVERALL RED RESULTS, BY KIND OF RED SHIP.  (ATTRITION IS PROPORTIONAL.)
    KIND OF RED SHIP                      1       2       3       4
    INITIAL RED SHIPS IN REGION        1.81    2.24   23.08   15.42
    RED SHIPS KILLED                    .01     .01     .01     .00
    RESULTANT RED SHIPS IN REGION      1.80    2.23   23.08   15.41
END OF SUBROUTINE SUBSUB
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE CTFMOD
ATT=      .0001

 UBAEWL     UBAEW
    .1400      .1400

 AEWD       STAR
   220.00     200.00

 CAPML      CAPM       BUCAP      OLIA       WVSIZ
     1.50       2.00       6.00        .67       2.00

 T1         T2         T3         T4
     8.00       3.00       0.00       1.00

 VCAP       CAPMR      TCAP       CAPSTAR
    12.00      50.00       1.50     100.00

 BARL
  1000.00

 UBASWL     BAREAL
    .1000   400.0000

 UBASW      BAREA
    .2500  1900.0000

 ASWF       PKASW      ST         PDIN        PKIN       ZLAMPF
    .0500      .5000     1.8048      .2000       .5000     2.0000

 PKIIN      ESR        ESLR       SUBSOR
      .15      15.00       8.00      15.00

 TPS
     2.50

 ZMPCAP     ZMPOLI
     3.00       3.00

 STG        ZMPSTG
     2.23       5.00

 STSALV
     2.00

 XASW       XAEW       XASWL      XAEWL
    20.00      20.00       0.00       0.00

 XEASWA     XEASWN     XEAAW
     0.00       0.00       0.00

 XPLAT
     2.00
```

```
   XD               .00
   XPDASW           .00
   XSURS1          1.80
   XSTG1           2.23
   XSURS2          1.80
   XTOTE           0.00
   XSURS3          1.80
   RS(1,L)=        1.8047 RS(2,L)=        2.2309 FOR L = 5
   XSALVS          3.61
   XPST             .10
   XAEWSTA         2.80
   XCAPSTA          .00
   XZ            316.50
   XZ            316.50

START OF ITERATION I= 1 THROUGH ATTRITION PORTION OF CTFMOD
ATT=      .0001
   XTB           -33.34  XTB CAN BE NEGATIVE.  IF SO, FUNCT1 WILL SET APPROPRIATE VALUES TO 0.
   XWB             0.00
   XDLI            0.00
   XDLI            0.00
   XDLIENG         0.00
   XL            -18.10   XL CAN BE NEGATIVE.  IF SO, FUNCT3 WILL SET APPROPRIATE VALUES TO 0.
   XTHETST         0.00
   XCAPENG         0.00
   XDLIENG         0.00
   XCAPENG         0.00
--------------------------
START SUBROUTINE ATRTIA
    FK=    0.0000     EK=     0.0000     BK=     0.0000
END OF SUBROUTINE ATRTIA
--------------------------
FOR KK= 1,  TATK=     0.0000
FOR KK= 2,  TATK=     0.0000
FOR KK= 3,  TATK=     0.0000
ATKT(1)=     0.0000 ATKT(2)=     0.0000 ATKT(3)=     0.0000 ATKT(
  AT(1)=      .0000   AT(2)=      .0000   AT(3)=      .0001   AT(
 FOR K= 1,  AT1(K)=      .0000 -- SO FAR (THROUGH K = 1) ATKTT=     0.0000
              AEST=    4.9069                          AESCKT=     0.0000
            FGHTRL=    0.0000                          FGHTRK=     0.0000
ATT=      .0001
   XTB           -10.02  XTB CAN BE NEGATIVE.  IF SO, FUNCT1 WILL SET APPROPRIATE VALUES TO 0.
   XWB             0.00
   XDLI            C.00
   XDLI            0.00
   XDLIENG         0.00
   XL            111.79   XL CAN BE NEGATIVE.  IF SO, FUNCT3 WILL SET APPROPRIATE VALUES TO 0.
   XTHETST         0.00
   XCAPENG         0.00
   XDLIENG         0.00
   XCAPENG         0.00
--------------------------
START SUBROUTINE ATRTIA
    FK=    0.0000     EK=     0.0000     BK=     0.0000
END OF SUBROUTINE ATRTIA
--------------------------
FOR KK= 2,  TATK=     0.0000
FOR KK= 3,  TATK=     0.0000
ATKT(1)=     0.0000 ATKT(2)=     0.0000 ATKT(3)=     0.0000 ATKT(
  AT(1)=      .0000   AT(2)=      .0000   AT(3)=      .0001   AT(
 FOR K= 2,  AT1(K)=      .0000 -- SO FAR (THROUGH K = 2) ATKTT=     0.0000
              AEST=    4.9059                          AESCKT=     0.0000
            FGHTPL=    0.0000                          FGHTRK=     0.0000
ATT=      .0001
   XTB             -.85  XTB CAN BE NEGATIVE.  IF SO, FUNCT1 WILL SET APPROPRIATE VALUES TO 0.
   XWB             0.00
```

```
   XDLI             0.00
   XDLI             0.00
   XDLIENG          0.00
   XL             171.79   XL CAN BE NEGATIVE.  IF SO, FUNCT3 WILL SET APPROPRIATE VALUES TO 0.
   XTHETST          2.07
   XCAPENG           .00
   XDLIENG          0.00
   XCAPENG           .00
-------------------------
START SUBROUTINE ATRTIA
    FK=      .0010     EK=      .0013     BK=      .0000
END OF SUBROUTINE ATRTIA
-------------------------
FOR KK= 3,  TATK=      .0000
ATKT(1)=     0.0000 ATKT(2)=     0.0000 ATKT(3)=      .0000 ATKT(
  AT(1)=      .0000   AT(2)=      .0000   AT(3)=      .0001   AT(
 FOR K= 3,  AT1(K)=      .0001 -- SO FAR (THROUGH K = 3) ATKTT=      .0000
                AEST=    4.9046                         AESCKT=      .0013
            FGHTRL=      .0010                          FGHTRK=      .0010
 AESCK=     .0013

DISPLAY RESULTS OF ASM-VS-SHIP BATTLE
   XATS1           11.15
FMRBT(1)=      .0000 FMRBT(2)=      .0000 FMRBT(3)=      .0000 FMRBT(
FMRBT(NKRB+1)=    1.0000

 PKSS       FPPL1      PKPL1      PKPL2      PKPLD      FPPL2
      .75       7.00        .75        .33        .50       1.00

 TAB10
   8.0000    16.0000    24.0000    32.0000    40.0000    48.0000    56.0000    64.0000
  72.0000    80.0000    88.0000    96.0000   104.0000   112.0000   120.0000   128.0000
 136.0000   144.0000   152.0000   160.0000

 TAB13
    .6000      .3000      .1500      .0750      .0350      .0150      .0100      .0050
   0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
   0.0000     0.0000     0.0000     0.0000
   XATS2           11.15
   XMPPLAT          5.58
   XATS3            5.58
   XATS4            5.57
 ENACD=     .0000  PIACD=      .0000   FACD=      .0000
 FDMCV=     .0000 XFGHTR=    26.0646
 ADMCV=     .0000 XATTCK=    35.6260
   XPSA              .02
   XEFFCM            .00
  PMAAW    0.0000   PMASW    0.0000   PMURG    0.0000   PTAAW    0.0000   PTASW    0.0000   PTURG    0.0000
 SURAAW    1.0000  SURASW    1.0000  SURURG    1.0000
  XEAAW    0.0000  XEASWA    0.0000  XEASWN    0.0000   XURGS    0.0000

START OF ITERATION I= 2 THROUGH ATTRITION PORTION OF CTFMOD
ATT=      .0001
   XTB             -6.68  XTB CAN BE NEGATIVE.  IF SO, FUNCT1 WILL SET APPROPRIATE VALUES TO 0.
   XWB              0.00
   XDLI             0.00
   XDLI             0.00
   XDLIENG          0.00
   XL             101.90   XL CAN BE NEGATIVE.  IF SO, FUNCT3 WILL SET APPROPRIATE VALUES TO 0.
   XTHETST           .28
   XCAPENG           .00
   XDLIENG          0.00
   XCAPENG           .00
-------------------------
START SUBROUTINE ATRTIA
    FK=      .0001     EK=      .0002     BK=      .0000
```

```
END OF SUBROUTINE ATRTIA
--------------------------
FOR KK= 1,  TATK=      .0000
FOR KK= 2,  TATK=      .0000
FOR KK= 3,  TATK=      .0000
ATKT(1)=      .0000 ATKT(2)=      .0000 ATKT(3)=      .0000 ATKT(
  AT(1)=      .0000   AT(2)=      .0000   AT(3)=      .0001   AT(
 FOR K= 1,  AT1(K)=      .0000 -- SO FAR (THROUGH K = 1) ATKTT=      .0000
              AEST=    4.9066                         AESCKT=      .0002
            FGHTPL=      .0001                        FGHTRK=      .0001
ATT=      .0001
   XTB            -.85  XTB CAN BE NEGATIVE.  IF SO, FUNCT1 WILL SET APPROPRIATE VALUES TO 0.
   XWB            0.00
   XDLI           0.00
   XDLI           0.00
   XDLIENG        0.00
   XL           171.79   XL CAN BE NEGATIVE.  IF SO, FUNCT3 WILL SET APPROPRIATE VALUES TO 0.
   XTHETST        2.07
   XCAPENG         .00
   XDLIENG        0.00
   XCAPENG         .00
--------------------------
START SUBROUTINE ATRTIA
    FK=     .0009     EK=      .0011     BK=      .0000
END OF SUBROUTINE ATRTIA
--------------------------
FOR KK= 2,  TATK=      .0000
FOR KK= 3,  TATK=      .0000
ATKT(1)=      .C000 ATKT(2)=      .0000 ATKT(3)=      .0000 ATKT(
  AT(1)=      .0000   AT(2)=      .0000   AT(3)=      .0001   AT(
 FOR K= 2,  AT1(K)=      .0000 -- SO FAR (THROUGH K = 2) ATKTT=      .0000
              AEST=    4.9046                         AESCKT=      .0013
            FGHTRL=      .0009                        FGHTRK=      .0010
ATT=      .0001
   XTB           17.48  XTB CAN BE NEGATIVE.  IF SO, FUNCT1 WILL SET APPROPRIATE VALUES TO 0.
   XWB            7.00
   XDLI          14.00
   XDLI           0.00
   XDLIENG        0.C0
   XL           291.79   XL CAN BE NEGATIVE.  IF SO, FUNCT3 WILL SET APPROPRIATE VALUES TO 0.
   XTHETST        6.28
   XCAPENG         .00
   XDLIENG        0.00
   XCAPENG         .00
--------------------------
START SUBROUTINE ATRTIA
    FK=     .0021     EK=      .0027     BK=      .0000
END OF SUBROUTINE ATRTIA
--------------------------
FOR KK= 3,  TATK=      .0000
ATKT(1)=      .0000 ATKT(2)=      .0000 ATKT(3)=      .0000 ATKT(
  AT(1)=      .0000   AT(2)=      .0000   AT(3)=      .0001   AT(
 FOR K= 3,  AT1(K)=      .0001 -- SO FAR (THROUGH K = 3) ATKTT=      .0000
              AEST=    4.8998                         AESCKT=      .0040
            FGHTRL=      .0021                        FGHTRK=      .0031
 AESCK=     .0C40

DISPLAY RESULTS OF AIR-TO-AIR  BATTLE
  AESC=     4.9029 XFGHTR=   26.0615
 AT(1)=     .0000
 AT(2)=     .0000
 AT(3)=     .0C01
 FTSORU=    0.0000  ATSORU=      .0128
ATABT(1,KRB)=      .4500 ATABT(2,KRB)=     1.2004 FOR KRB = 1
ATABT(1,KRB)=      .2048 ATABT(2,KRB)=      .4286 FOR KRB = 2
ATABT(1,KRB)=      .0286 ATABT(2,KRB)=      .5220 FOR KRB = 3
```

```
   AESCAB(1)=    3.9963    AESCAB(2)=   31.5359
RELATIVE CARRIER CAPABILITY (XEFFCM) IS NOW  .0000.
END OF SUBROUTINE CTFMOD
--------------------------------------------------

REMINDER--THIS IS PERIOD   9, TASK FORCE IS IN REGION  5.

--------------------------------------------------
START SUBROUTINE SHPSHP
BLUE AIRCRAFT FROM CARRIER--     .00.  BLUE AIRCRAFT REQUIRED TO DESTROY ALL VULNERABLE RED SHIPS--  973.27.
 XATTCK=   35.6260,  AAK=      .0000,  ATSORU=     .0129
 XFGHTR=   26.0615,  FAK=     0.0000,  FTSORU=    0.0000
BLUE SURFACE SHIP RESULTS
    KIND OF BLUE SHIP                       XPLAT  XEAAW XEASWA XEASWN  XURGS
    INITIAL BLUE SHIPS IN TASK FORCE         2.00   0.00   0.00   0.00   0.00
    BLUE SHIPS DESTROYED                     0.00   0.00   0.00   0.00   0.00
    RESULTANT BLUE SHIPS IN TASK FORCE       2.00   0.00   0.00   0.00   0.00
    CARRIER CAPABILITY DEGRADED BY  .7258, NEW RELATIVE CARRIER CAPABILITY (XEFFCM) EQUALS  .0000.
RED SURFACE SHIP RESULTS (NOTE--RED SHIP KINDS 1 AND 2 ARE SUBMARINES, WHICH DO NOT PARTICIPATE IN THIS INTERACTION.)
    KIND OF RED SHIP                            3      4
    INITIAL RED SHIPS IN REGION             23.08  15.41
    RED SHIPS VULNERABLE TO BLUE ATTACK     17.31  10.79
    RED SHIPS DESTROYED BY BLUE AIRCRAFT      .00    .00
    RED SHIPS DESTROYED BY BLUE SHIPS        0.00   0.00
    RESULTANT RED SHIPS IN REGION           23.08  15.41
 ENACD=    1.1160  PIACD=     .0093   FACD=    0.0000
 FDMCV=    0.0000 XFGHTR=   26.0615
 ADMCV=    0.0000 XATTCK=   35.6260
END OF SUBROUTINE SHPSHP
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE POWERP
TASK FORCE IS IN REGION  5.
NO BLUE AIRCRAFT AVAILABLE.  NO POWER PROJECTION PERFORMED.
END OF SUBROUTINE POWERP
--------------------------------------------------

--------------------------------------------------
START SUBROUTINE ADDMOE
 XPLAT=    2.0000 XEFFCM=     .0000  XSHIP=    2.0000  ISTOP= 1    ITP=   9
END OF SUBROUTINE ADDMOE
--------------------------------------------------

END OF PERIOD    9

END OF PROGRAM MEDMOD
```

SUMMARY OF RESULTS OF MEDMOD SIMULATION

NOTE--1)PERIOD -1 CORRESPONDS TO INITIAL VALUES.
2)PERIOD 0 CORRESPONDS TO VALUES AFTER THE D-DAY SHOOTOUT.
3)ALL OTHER VALUES LISTED BELOW ARE AS OF THE END OF THE CORRESPONDING PERIOD.
4)NUMBER OF CARRIERS (XPLAT) IS 2.0. TOTAL BLUE SURFACE SHIPS COLUMN BELOW EXCLUDES CARRIERS.

	SUMMARY OF RESULTS FOR BLUE									SUMMARY OF RESULTS FOR RED				
ITP	XEFFCM	CUMLTV WGHTED EFCTVNS	TOTAL SURFACE SHIPS	TOTAL SUBS	FIGHTER AIRCPAFT (ON CV)	ATTACK AIRCRAFT (ON CV)	LAND-BSD INTCPTR AIRCRAFT	POWER PROJECTN SORTIES	CUMLTIVE WGHTD PP SORTIES	TOTAL SURFACE SHIPS	TOTAL SUBS	TOTAL BOMBERS	FIGHTER AIRCRAFT	INTCPTR AIRCRAFT
-1	1.0000	0.0000	20.00	24.00	48.00	72.00	246.00	0.00	0.00	86.00	24.60	160.00	200.00	50.00
0	.9560	0.0000	19.85	24.00	47.92	71.88	246.00	0.00	0.00	84.00	22.60	160.00	200.00	50.00
1	.9557	0.0000	19.85	9.91	47.92	71.88	246.00	0.00	0.00	80.23	21.77	160.00	200.00	50.00
2	.9553	0.0000	19.82	8.56	47.92	70.42	246.00	0.00	0.00	76.02	21.43	160.00	200.00	50.00
3	.8457	.8457	14.94	6.90	42.50	62.21	246.00	0.00	0.00	67.30	18.53	152.48	196.85	36.06
4	.8195	1.6652	13.42	6.06	35.21	54.44	235.66	.05	.01	63.00	17.78	118.10	163.50	32.69
5	.8018	2.4669	12.36	5.73	28.79	47.85	235.66	.05	.03	60.92	17.42	111.62	160.81	28.65
6	.5246	2.9915	1.60	5.25	28.79	41.45	231.20	0.00	.03	59.13	16.01	51.15	113.55	26.67
7	.2804	3.2719	.09	5.21	28.79	36.30	231.20	0.00	.03	58.08	15.71	50.85	112.67	25.40
8	.0004	3.2723	0.00	5.20	26.06	35.63	205.31	0.00	.03	58.06	15.40	14.31	67.78	25.40
9	.0000	3.2723	0.00	5.19	26.06	35.63	187.61	0.00	.03	58.06	15.39	2.83	35.53	25.40

THE DATA BELOW GIVE THE LOCATION OF THE TASK FORCE (LOCTF) AT THE START OF EACH TIME PERIOD (ITP) FOR ALL ITP, SIMULATED OR NOT

ITP =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
LOCTF =	1	1	2	3	3	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

END OF SUMMARY TABLE

APPENDIX E

THE MEDMOD COMPUTER PROGRAM

THE MEDMOD COMPUTER PROGRAM

The following is a copy of the code of the MEDMOD computer program. Copies of this program on appropriate media (cards, tape, etc.) can be obtained, with the appropriate approvals, from the Institute for Defense Analyses. Also available from IDA, with the appropriate approvals, are copies of hypothetical inputs on appropriate media, and copies of the printouts produced by this program with these inputs.

PROGRAM DRIVER

```
C* DECK DRIVER
      OVERLAY(DVER,0,0)
      PROGRAM DRIVER(INPUT,OUTPUT,TAPE6=OUTPUT,TAPE15,TAPE16,TAPE10)
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,YURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C*
```

```
C* COMDECK COMIGO
      COMMON/COMIGO/ IGO
C*
C
C* COMDECK COMOUT
      COMMON/COMOUT/ CWPPAS,CWTPTF,PPSORT,NTPSIM,LTASKF(90)
C*
C
      DIMENSION IHEAD(3,28)
      DATA IHYP/4H----/
      DATA IHEAD   /4H    ,4H    ,4H  XE,4H    ,4H    ,4HFFCM,
     14H  CU,4H  WG,4H  XE,4HMLTV,4HHTED,4HFFCM,4H   T,4H SUR,4H   S,
     24HOTAL,4HFACE,4HHIPS,4H    ,4H   T,4H    ,4H    ,4HOTAL,4HSUBS,
     34H FIG,4HAIRC,4H (ON,4HHTER,4HRAFT,4H CV),4H  AT,4HAIRC,4H (ON,
     44HTACK,4HRAFT,4H CV),4HLAND,4H INT,4HAIRC,4H-BSD,4HCPTR,4HRAFT,
     54H   P,4HPROJ,4H SOR,4HOWER,4HECTN,4HTIES,4HCUML,4HWGHT,4H SOR,
     64HTIVE,4HD PP,4HTIES,4H   T,4H SUR,4H   S,4HOTAL,4HFACE,4HHIPS,
     74H    ,4H   T,4H    ,4HOTAL,4HSUBS,4H    ,4H   T,4H BOM,
     84H    ,4HOTAL,4HBERS,4H    ,4H FIG,4HAIRC,4H    ,4HHTER,4HRAFT,
     94H    ,4H INT,4HAIRC,4H    ,4HCPTR,4HRAFT/
C
      IHEAD(3,3)=4H EFC
      IHEAD(3,4)=4HTVNS
C
 1001 FORMAT(40H1SUMMARY OF RESULTS OF MEDMOD SIMULATION )
 1010 FORMAT(1H+,6HNOTE--  )
 1011 FORMAT(1H0,6X,42H1)PERIOD -1 CORRESPONDS TO INITIAL VALUES.  )
 1012 FORMAT(1H ,6X,59H2)PERIOD  0 CORRESPONDS TO VALUES AFTER THE D-DAY
     1 SHOOTOUT. )
 1013 FORMAT(1H ,6X,78H3)ALL OTHER VALUES LISTED BELOW ARE AS OF THE END
     1 OF THE CORRESPONDING PERIOD. )
 1014 FORMAT(1H ,6X,31H4)NUMBER OF CARRIERS (XPLAT) IS,F6.1,59H.  TOTAL
     1BLUE SURFACE SHIPS COLUMN BELOW EXCLUDES CARRIERS. )
 1020 FORMAT(1H //1H ,5X,6A4,27HSUMMARY OF RESULTS FOR BLUE,6A4,5X,2A4,
     126HSUMMARY OF RESULTS FOR RED,2A4)
 1021 FORMAT(1H ,3X,8A4,5(1X,2A4),4X,4A4,3(1X,2A4))
 1022 FORMAT(1H+,3HITP )
      MST=10
C
      PRINT 10
   10 FORMAT(45H1INPUTS FOR THIS RUN OF MEDMOD ARE AS FOLLOWS)
C*         OVERLAY 1 IS PROGRAM INP
      CALL OVERLAY(4LOVER,1,0,6HRECALL)
C*
      WRITE(MST,1001)
      WRITE(MST,1011)
      WRITE(MST,1010)
      WRITE(MST,1012)
      WRITE(MST,1013)
      WRITE(MST,1014)   XPLAT
      WRITE(MST,1020) (IHYP,J=1,16)
      DO 50 IROW=1,3
      WRITE(MST,1021) (IHEAD(IROW,J),J=1,28 )
   50 CONTINUE
      WRITE(MST,1022)
C
      PRINT 20
```

```
   20 FORMAT(47H1OUTPUTS FROM THIS RUN OF MEDMOD ARE AS FOLLOWS)
C*          OVERLAY 2 IS PROGRAM MEDMOD
      CALL OVERLAY(4LOVER,2,0,6HRECALL)
C*
      WRITE(MST,1079)
      WRITE(MST,1080)
      ICOL = 1
      MCOL = MIN0(MAXTP,30)
      WRITE(MST,1081) (ITP,ITP=ICOL,MCOL)
      WRITE(MST,1082) (LTASKF(ITP),ITP=ICOL,MCOL)
      IF(MAXTP.LE.30) GO TO 30
      ICOL = 31
      MCOL = MIN0(MAXTP,60)
      WRITE(MST,1081) (ITP,ITP=ICOL,MCOL)
      WRITE(MST,1082) (LTASKF(ITP),ITP=ICOL,MCOL)
      IF(MAXTP.LE.60) GO TO 30
      ICOL = 61
      MCOL = MIN0(MAXTP,90)
      WRITE(MST,1081) (ITP,ITP=ICOL,MCOL)
      WRITE(MST,1082) (LTASKF(ITP),ITP=ICOL,MCOL)
      IF(MAXTP.LE.90) GO TO 30
      WRITE(MST,1083)
 1079 FORMAT(1H0)
 1080 FORMAT(128HOTHE DATA BELOW GIVE THE LOCATION OF THE TASK FORCE (LO
     1CTF) AT THE START OF EACH TIME PERIOD (ITP) FOR ALL ITP, SIMULATED
     2 OR NOT)
 1081 FORMAT(9H0 ITP    =,30I4)
 1082 FORMAT(9H  LOCTF =,30I4)
 1083 FORMAT(44HONOTE--MAXTP EXCEEDS THE DIMENSION OF LTASKF)
   30 CONTINUE
      WRITE(MST,1079)
C*
      WRITE(MST,1090)
      WRITE(MST,1099)
 1090 FORMAT(21HOEND OF SUMMARY TABLE )
 1099 FORMAT(36HOEND OF THIS RUN OF THE MEDMOD MODEL )
      STOP 6400
      END
```

PROGRAM MEDMOD

```
C* DECK MEDMOD
      OVERLAY(OVER,2,0)
      PROGRAM MEDMOD
C*
C*    MEDITERRANEAN MODEL -- MEDMOD
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
```

PROGRAM MEDMOD

```
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C*
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C* COMDECK COMGA
      COMMON/COMGA/ NTPSLA,BMR(2,3),ESC(2)
C*
C*
C* COMDECK COMIGO
      COMMON/COMIGO/ IGO
C*
C*
C* COMDECK COMSOR
      COMMON/COMSOR/ FTSORU,ATSORU
C*
C*
C* COMDECK COMOUT
      COMMON/COMOUT/ CWPPAS,CWTPTF,PPSORT,NTPSIM,LTASKF(90)
C*
C*
      PRINT 1
    1 FORMAT(24HOSTART PROGRAM MEDMOD
C*
C*    RECORD INITIAL (INPUT) VALUES OF SELECTED RESOURCES
C     AND INITIALIZE SELECTED PARAMETERS
C*
      FGHTRI = XFGHTR
      ATTCKI = XATTCK
      ATSORU = 0.
      FTSORU = 0.
      CWTPTF = 0.
      CWPPAS = 0.
      XEFFCM = 1.
      XCAPST = 0.
      NTPSLA = 0
      NTPSIM = 0
      PPSORT = 0.
      LOCTF = LTFMP(1)
C*
      CALL PRTSUM(LOCTF,-1)
C*
C*    DDAY MODELS THE DDAY SHOOTOUT
C*
      IF(LOCTF .GT. 0) CALL DDAY(LOCTF)
C*
      CALL PRTSUM(LOCTF,0)
C*
      IF(XEFFCM.LE.0.) GO TO 3000
C*
      DO 2000 ITP=1,MAXTP
C*
      PRINT 9
    9 FORMAT(120HO-----------------------------------------------------------
     1------------------------------------------------------------)
      PRINT 9
      PRINT 10, ITP
```

```
   10 FORMAT(16HOSTART OF PERIOD,I5)
C*
      IF(IGO.EQ.ITP) CALL TIMET(ITP)
C*
      PRINT 11, LOCTF
   11 FORMAT(8HO LOCTF=,I2 )
      LTASKF(ITP) = LOCTF
C*
      IF(LOCTF.EQ.0) GO TO 1000
C*
C*    GNAATK GENERATES AIR ATTACKS ON THE TASK FORCE
C*
      CALL GNAATK(LOCTF,ITP)
C*
C*    PLBAB MODELS THE ATTEMPT BY THE RED AIR ATTACK TO PENETRATE THE
C*    BLUE LAND-BASED AIR BARRIER
C*
      CALL PLBAB(LOCTF)
C*
C*    SUBSUB MODELS BLUE SUBMARINES IN DIRECT SUPPORT OF THE TASK FORCE
C*    VERSUS RED SUBMARINES AND RED SURFACE SHIPS IN THE SAME LOCATION
C*
      CALL SUBSUB(LOCTF)
C*
C*    CTFMOD EXERCISES THE CTF MODEL BASED ON IDA REPORT R-245
C*
      CALL CTFMOD(LOCTF)
C*
C*    SHPSHP MODELS SURFACE SHIP VS SURFACE SHIP WARFARE
C*
      CALL SHPSHP(LOCTF,ITP)
C*
C*    POWERP CALCULATES POWER PROJECTION RESULTS
C*
      CALL POWERP(LOCTF,ITP)
C*
 1000 CONTINUE
C*
C*    ADDMOE DETERMINES WHETHER TO STOP THE SIMULATION
C*
      CALL ADDMOE(ITP,ISTOP)
C*
      IF(ISTOP.EQ.1) GO TO 1900
C*
C*    MOVTF MOVES THE TASK FORCE (IF APPROPRIATE) TO A NEW AREA AND
C*    ASSESS ANY SUB-BARRIER ATTRITION TO THE TASK FORCE
C*
      CALL MOVTF(LOCTF,ITP)
C*
C*    MOVRS MOVES RED SURFACE SHIPS AND RED SUBMARINES AND ASSESSES
C*    ANY SUB-BARRIER ATTRITION TO THESE SHIPS AND SUBMARINES
C*
      CALL MOVRS(LOCTF,ITP)
C*
C*    ABATCK MODELS BLUE AIR ATTACKS ON RED AIRBASES
C*
      CALL ABATCK(LOCTF)
```

PROGRAM MEDMOD

```
C*
 1900 CONTINUE
C*
      CALL PRTRES(LOCTF,ITP)
C*
      CALL PRTSUM(LOCTF,ITP)
C*
      PRINT 1990, ITP
 1990 FORMAT(14H0END OF PERIOD,I5)
C*
      IF(ISTOP.EQ.1) GO TO 3000
C*
 2000 CONTINUE
C*
      NTPSIM = MAXTP
      GO TO 3020
C*
 3000 CONTINUE
C*
      NTPSIM=ITP
      NTPSP1 = NTPSIM + 1
      DO 3010 ITP=NTPSP1,MAXTP
 3010 LTASKF(ITP) = LOCTFF(ITP,LGTHMP,LTFMP,MIMP)
 3020 CONTINUE
C*
      WRITE(6,2)
    2 FORMAT(22H0END OF PROGRAM MEDMOD   )
      END
```

FUNCTION LOCTFF

```
C* DECK LOCTFF
      FUNCTION LOCTFF(ITP,LGTHMP,LTFMP,MIMP)
      DIMENSION LGTHMP(MIMP),LTFMP(MIMP)
C*
C*
      LENGTH = 0
      DO 10 IMP=1,MIMP
      LENGTH = LENGTH + LGTHMP(IMP)
      IF(ITP.GT.LENGTH) GO TO 10
      LOCTF = LTFMP(IMP)
      GO TO 20
   10 CONTINUE
      LOCTF = LTFMP(MIMP)
   20 CONTINUE
      LOCTFF = LOCTF
C
      RETURN
      END
```

SUBROUTINE ABATCK

```
C* DECK ABATCK
      SUBROUTINE ABATCK(L)
C*
C*    ABATCK MODELS BLUE AIR ATTACKS ON RED AIRBASES
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJD(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJD(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSDRR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE ABATCK

```
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C* COMDECK COMGA
      COMMON/COMGA/ NTPSLA,BMR(2,3),ESC(2)
C*
C* COMDECK COMSOR
      COMMON/COMSOR/ FTSORU,ATSORU
C*
C*
      REAL ID,IDK
      DIMENSION SA(2),SD(2),SFE(2),SAA(2),SAH(2),SAK(2),SDA(2),SDH(2),
     1SDK(2),SEA(2),SEH(2),SEK(2)
      DIMENSION PKVED(2),PKVDE(2),PKVDA(4),PKVAD(4),VPSA(4),VPKS(4)
      DIMENSION RSAMK(2),SAL(2),ABATKR(2)
      DIMENSION A(5),AOB(5),AOBS(5),AOBN(5),AOBSK(5),AOBNK(5),ANEW(5)
C
      WRITE(6,1)
      WRITE(6, 501)
    1 FORMAT(51H0--------------------------------------------------)
  501 FORMAT(24H START SUBROUTINE ABATCK)
      WRITE(6,502) L
  502 FORMAT(28H TASK FORCE IS NOW IN REGION  ,I3)
C
      FTSORU=0.
      ATSORU=0.
C
      IF(L.EQ.0) GO TO 98
C
      BMRT = 0.
      VRAT = 0.
      DO 7 KRB=1,NKRB
      BMRT = BMRT + ATABT(1,KRB) + ATABT(2,KRB)
      VRAT = VRAT + ATABT(1,KRB)
    7 CONTINUE
      VRAT = VRAT + AESCAB(1)
      RACT = BMRT + AESCAB(1) + AESCAB(2)
      IF(RARBAB(1).GE.BMRT) GO TO 95
      IF(RARBAB(2).GE.RACT) GO TO 95
      IF(RARBAB(3).GE.VRAT) GO TO 95
C
C  BLUE AIRCRAFT FROM CARRIER
C
      AA=AMIN1(XATTCK,ATTCKI*XEFFCM)*FAACA(L)
      F1=AMIN1(XFGHTR,FGHTRI*XEFFCM)
      F1=AMAX1(0.,F1-BUCAP*XCAPST)
      FA=F1*FFACA(L)
      FE=F1*FFACE(L)
      SFE(1)=FE*AASRFE(L)
      SA(1)=AA*AASRAA(L)
      SA(2)=FA*AASRFA(L)
      IATTCK=1
      IF(SA(1)+SA(2).LT.XIA(L)) IATTCK=0
      IF(SFE(1).LT.XIE(L) .AND. IATTCK.GE.1) IATTCK=-1
      IF(IATTCK.GE.1) GO TO 5
      SA(1)=0.
      SA(2)=0.
```

```
      SFE(1)=0.
    5 CONTINUE
      ASRAA=AMAX1(1.,AASRAA(L))
      ASRFA=AMAX1(1.,AASRFA(L))
      ASRFE=AMAX1(1.,AASRFE(L))
      FTSORU=PFFCNF*(SA(2)/ASRFA+SFE(1)/ASRFE)
      ATSORU=PAFCNF*SA(1)/ASRAA
      IF(IATTCK) 97,96,6
    6 CONTINUE
C
C  RED DEFENDERS
C
      IF(NTPSLA .GT. 0 .AND. NTPSLA .LT. IATKRT(L)) GO TO 4
      IATF=1
      GO TO 3
    4 IATF=2
    3 FESCAD=AVALED(L,IATF)
      ED=AESCAB(1)*FESCAD
      ID=AINTCT
      SD(1)=ED*AASRED
      SD(2)=ID*AASRID
C
C  AIR-TO-AIR INTERACTION--BLUE ATTACKERS, RED DEFENDERS
C
      WRITE(6,505)
  505 FORMAT(54H NOTE--AIRAIR AND ATRTSS DEAL WITH NUMBERS OF SORTIES.)
C  CONVERT KILL PROBABILITY MATRICES TO VECTORS
      IKE=1
      DO 10 IKD=1,2
      IADD=(IKD-1)+IKE
      PKVED(IADD)=AAPKED(IKE,IKD)
      PKVDE(IKD)=AAPKDE(IKD,IKE)
   10 CONTINUE
      DO 11 IKA=1,2
      DO 11 IKD=1,2
      IADD=(IKD-1)*2+IKA
      PKVAD(IADD)=AAPKAD(IKA,IKD)
      IADD=(IKA-1)*2+IKD
      PKVDA(IADD)=AAPKDA(IKD,IKA)
   11 CONTINUE
      CALL AIRAIR(SFE,SD,SA,AAPDED,AAPDDE,AAPDDA,PKVED,PKVDE,PKVDA,
     1PKVAD,AAPAJD,AAAEED,AAAEDE,AAAEDA,AACA,1,2,2,IAAED,IAADA,
     2SEA,SEK,SEH,SDA,SDK,SDH,SAA,SAK,SAH)
C  SAA WILL ENTER THE SAM ROUTINE
      AAK=SAK(1)/ASRAA
      FAK=SAK(2)/ASRFA
      FEK=SEK(1)/ASRFE
      FTSORU = FTSORU - PFFCNF*(FAK+FEK)
      ATSORU = ATSORU - PAFCNF*AAK
      EDK=SDK(1)/AMAX1(1.,AASRED)
      IDK=SDK(2)/AMAX1(1.,AASRID)
      WRITE(6,526)
      WRITE(6,527)
      WRITE(6,528) AA,FA,FE,ED,ID
      WRITE(6,525) SA(1),SA(2),SFE(1),SD(1),SD(2)
      WRITE(6,529) AAK,FAK,FEK,EDK,IDK
  525 FORMAT(27H SORTIES (AIRCRAFT FLYING)  ,5(6X,F10.3))
```

SUBROUTINE ABATCK

```
  526 FORMAT(1H ,33X,73HBLUE ATT-     BLUE FIGHTER    BLUE FIGHTER      R
     1ED ESCORT      RED INTCPTR )
  527 FORMAT(1H ,30X,76HACK AIRCRAFT    A/C ON ATTACK    A/C ON ESCORT   A/
     1C ON DEFENSE   A/C ON DEFENSE )
  528 FORMAT(1H ,16HINITIAL AIRCRAFT ,10X,5(6X,F10.3))
  529 FORMAT(27H AIRCRAFT KILLED AIR-TO-AIR ,5(6X,F10.3))
C
C  BLUE ATTACKERS/RED SAMS INTERACTION
C
C  CONVERT MATRICES TO VECTORS FOR APPROPRIATE ATRTSS INPUTS
      DO 20 KBAC=1,2
      DO 20 KSAM=1,NABSAM
      IND1=(KBAC-1)*NABSAM+KSAM
      IND2=(KSAM-1)*2 + KBAC
      VPKS(IND1)=ABPKS(KSAM,KBAC)
      VPSA(IND2)=ABPSA(KBAC,KSAM)
   20 CONTINUE
      CALL ATRTSS(ABRSAM,ABVGSS,SAA,ABPDA,VPSA,ABPKA,ABAVLS,ABANM,ABPDS,
     1VPKS,ABFASS,ABCAS,NABSAM,2,ABESGS,ABFSM,ABFVS,ABTSC,IABAF,IABAW,
     2SEA,SEH,RSAMK,ABATKR,SAH,SAL)
      IF(AASRAA(L) .LE. 1.) GO TO 23
      IF(SAL(1) .LE. 0.) GO TO 22
      AAL=(1.-(1.-SAL(1)/SAA(1))**AASRAA(L))*SAA(1)/AASRAA(L)
      GO TO 25
   22 AAL=0.
      GO TO 25
   23 AAL=SAL(1)
   25 IF(AASRFA(L) .LE. 1.) GO TO 28
      IF(SAL(2) .LE. 0.) GO TO 27
      FAL=(1.-(1.-SAL(2)/SAA(2))**AASRFA(L))*SAA(2)/AASRFA(L)
      GO TO 30
   27 FAL=0.
      GO TO 30
   28 FAL=SAL(2)
   30 CONTINUE
      FTSORU = FTSORU - PFFCNF*FAL
      ATSORU = ATSORU - PAFCNF*AAL
      WRITE(6,530) AAL,FAL
      WRITE(6,531) ABATKR(1),ABATKR(2)
  530 FORMAT(33H0BLUE AIRCRAFT LOST TO RED SAMS--,F10.3,17H ATTACK AIRCR
     1AFT,,F10.3,18H FIGHTER AIRCRAFT. )
  531 FORMAT(48H BLUE SORTIES ATTACKING VULNERABLE RED AIRBASE--,F10.3,
     120H BY ATTACK AIRCRAFT,,F10.3,21H BY FIGHTER AIRCRAFT. )
C
C  UPDATE RED SAM AND BLUE AIRCRAFT INVENTORIES
C
      DO 40 KSAM=1,NABSAM
      ABRSAM(KSAM)=ABRSAM(KSAM)-RSAMK(KSAM)
   40 CONTINUE
      XATTCK=XATTCK-AAK-AAL
      XFGHTR=XFGHTR-FAK-FAL-FEK
      XATTCK=AMAX1(XATTCK,0.)
      XFGHTR=AMAX1(XFGHTR,0.)
C
C  RED AIRCRAFT ON BASE--PRIORITY SHELTERING IS USED
C
      NKRA=NKRB+2
```

SUBROUTINE ABATCK

```
      NKRAM1=NKRA-1
      DO 60 KRA=1,NKRB
      A(KRA)=ATABT(1,KRA)
   60 CONTINUE
      A(NKRAM1)=AESCAB(1)-EDK
      A(NKRA)=AINTCT-IDK
      SHELA=SHEL
      TAOBS=0.
      TAOBN=0.
      DO 61 KRA=1,NKRA
      AOB(KRA)=A(KRA)*FACOB(KRA,IATF)
      AOBS(KRA)=AMIN1(AOB(KRA)*IKRAS(KRA),SHELA)
      AOBN(KRA)=AOB(KRA)-AOBS(KRA)
      AOBSK(KRA)=0.
      AOBNK(KRA)=0.
      TAOBS=TAOBS+AOBS(KRA)
      TAOBN=TAOBN+AOBN(KRA)
      SHELA=AMAX1(0.,SHELA-IKRAS(KRA)*A(KRA))
   61 CONTINUE
      TAOBS1=TAOBS/XNRAB
      TAOBN1=TAOBN/XNRAB
      SHEL1=SHEL/XNRAB
      ABATKR(1)=ABATKR(1)/XNRAB
      ABATKR(2)=ABATKR(2)/XNRAB
      SHELK=0.
      IF(ABATKR(1)+ABATKR(2) .LE. 0.) GO TO 66
      CALL ATRTAB(ABATKR,SHEL1,TAOBS1,TAOBN1,PARK,PBDRS,PBDRN,FHSK,
     1PBKRS,PBKRN,PASS,2,IABAEQ,SHELK,ASK,ANK)
      ASK=ASK*XNRAB
      ANK=ANK*XNRAB
      SHELK=SHELK*XNRAB
      IF(TAOBS .LE. 0.) GO TO 64
      DO 63 KRA=1,NKRA
      AOBSK(KRA)=ASK*AOBS(KRA)/TAOBS
   63 CONTINUE
   64 IF(TAOBN .LE. 0.) GO TO 66
      DO 65 KRA=1,NKRA
      AOBNK(KRA)=ANK*AOBN(KRA)/TAOBN
   65 CONTINUE
   66 SHELR=SHEL-SHELK
      DO 67 KRA=1,NKRA
      ANEW(KRA)=A(KRA)-AOBSK(KRA)-AOBNK(KRA)
   67 CONTINUE
      DO 68 KRA=1,NKRB
      ATABT(1,KRA)=ANEW(KRA)
   68 CONTINUE
      AESCAB(1)=ANEW(NKRAM1)
      AINTCT=ANEW(NKRA)
C
C     PRINT OUT AIRCRAFT DESTROYED ON GROUND
C
      IF(NKRB .GE. 2) GO TO 69
      WRITE(6,568)
      GO TO 70
   69 WRITE(6,569)
   70 WRITE(6,570)
      IHOLA=3HBOM
```

```
      IHOLB=3HBER
      WRITE(6,571) (IHOLA,IHOLB,I,I=1,NKRB)
      GO TO (72,73,74,75,76,77,78),NKRAM1
   72 WRITE(6,572)
      GO TO 80
   73 WRITE(6,573)
      GO TO 80
   74 WRITE(6,574)
      GO TO 80
   75 WRITE(6,575)
      GO TO 80
   76 WRITE(6,576)
      GO TO 80
   77 WRITE(6,577)
      GO TO 80
   78 WRITE(6,578)
   80 CONTINUE
  568 FORMAT(1H0,47X,20HKIND OF RED AIRCRAFT  )
  569 FORMAT(1H0,45X,39HK I N D   O F   R E D   A I R C R A F T   )
  570 FORMAT(18H  Q U A N T I T Y ,17X,6HSYMBOL )
  571 FORMAT(1H+,40X,6(3X,2A3,I1))
  572 FORMAT(1H+, 40X,20H    ESCORT   INTCPTR  )
  573 FORMAT(1H+, 50X,20H    ESCORT   INTCPTR  )
  574 FORMAT(1H+, 60X,20H    ESCORT   INTCPTR  )
  575 FORMAT(1H+, 70X,20H    ESCORT   INTCPTR  )
  576 FORMAT(1H+, 80X,20H    ESCORT   INTCPTR  )
  577 FORMAT(1H+, 90X,20H    ESCORT   INTCPTR  )
  578 FORMAT(1H+,100X,20H    ESCORT   INTCPTR  )
      WRITE(6,581) (    A(K),K=1,NKRA)
      WRITE(6,582) (  AOB(K),K=1,NKRA)
      WRITE(6,583) ( AOBS(K),K=1,NKRA)
      WRITE(6,584) (AOBSK(K),K=1,NKRA)
      WRITE(6,585) ( AOBN(K),K=1,NKRA)
      WRITE(6,586) (AOBNK(K),K=1,NKRA)
      WRITE(6,587) ( ANEW(K),K=1,NKRA)
      WRITE(6,588) SHEL,SHELK,SHELR
  581 FORMAT(41H INITIAL NUMBER OF AIRCRAFT                A ,8F10.3)
  582 FORMAT(41H AIRCRAFT ON BASE                        AOB ,8F10.3)
  583 FORMAT(41H SHELTERED AIRCRAFT                     AOBS ,8F10.3)
  584 FORMAT(41H SHELTERED AIRCRAFT KILLED             AOBSK ,8F10.3)
  585 FORMAT(41H NON-SHELTERED AIRCRAFT                 AOBN ,8F10.3)
  586 FORMAT(41H NON-SHELTERED AIRCRAFT KILLED         AOBNK ,8F10.3)
  587 FORMAT(41H RESULTANT NUMBER OF AIRCRAFT           ANEW ,8F10.3)
  588 FORMAT(1H0,10HSHELTERS--,F8.2,15H INITIALLY LESS,F8.2,17H DESTROYE
     1D YIELDS,F8.2,11H SURVIVING. )
      SHEL=SHELR
      PRINT  589, FTSORU,ATSORU
  589 FORMAT(9H  FTSORU=,F10.4,9H  ATSORU=,F10.4)
      GO TO 99
   95 WRITE(6,595)
  595 FORMAT(58H INSUFFICIENT RED AIRCRAFT TO MERIT A BLUE AIRBASE ATTAC
     1K.)
      GO TO 99
   96 WRITE(6,596)
  596 FORMAT(54H INSUFFICIENT BLUE ATTACK AIRCRAFT--NO AIRBASE ATTACK.)
      GO TO 99
   97 WRITE(6,597)
```

```
  597 FORMAT(54H INSUFFICIENT BLUE ESCORT AIRCRAFT--NO AIRBASE ATTACK.)
      GO TO 99
   98 WRITE(6,598)
  598 FORMAT(47H TASK FORCE IS IN REGION 0.  NO AIRBASE ATTACK.)
C
   99 WRITE(6,599)
      WRITE(6,2)
  599 FORMAT(25H END OF SUBROUTINE ABATCK)
    2 FORMAT(51H --------------------------------------------------)
C
      RETURN
      END
```

SUBROUTINE ADDMOE

```
C* DECK ADDMOE
      SUBROUTINE ADDMOE(ITP,ISTOP)
C*
C*    ADDMOE ADDS UP MOES AND DETERMINES WHETHER TO STOP THE SIMULATION
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACDB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSDRR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE ADDMOE

```
C*
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C*
C* COMDECK COMOUT
      COMMON/COMOUT/ CWPPAS,CWTPTF,PPSORT,NTPSIM,LTASKF(90)
C*
      WRITE(6,1)
      WRITE(6, 501)
    1 FORMAT(51H0--------------------------------------------------)
  501 FORMAT(24H START SUBROUTINE ADDMOE)
C*
      ISTOP = 0
C*
      IF(XPLAT .GT. 0. .AND. XEFFCM .LT. .00005) ISTOP=1
      XSHIP = XPLAT + XEAAW + XEASWA + XEASWN + XURGS
      IF(XSHIP.LE.0.) ISTOP=1
      IF(ISTOP.EQ.1) NTPSIM = ITP
C*
      PRINT 510, XPLAT,XEFFCM,XSHIP,ISTOP,ITP
  510 FORMAT(8H  XPLAT=,F10.4,8H XEFFCM=,F10.4,8H  XSHIP=,F10.4,8H  ISTO
     1P=,I2,8H    ITP=,I4)
C*
      WRITE(6, 599)
      WRITE(6,2)
  599 FORMAT(25H END OF SUBROUTINE ADDMOE)
    2 FORMAT(51H --------------------------------------------------)
C
C*
      RETURN
      END
```

SUBROUTINE AIRAIR

```
C* DECK AIRAIR
      SUBROUTINE AIRAIR(E,D,A,PDED,PDDE,PDDA,PKED,PKDE,PKDA,PKAD,PAJD,
     X    AEED,AEDE,AEDA,CA,NKE,NKD,NKA,IAFED,IAFDA,EA,EK,EH,DA,DK,DH,
     X    AA,AK,AH)
C*
C*    AIRAIR COMPUTES AIR-TO-AIR ATTRITION FOR ESCORTS VS DEFENDERS
C*           AND THEN FOR DEFENDERS VS ATTACKERS
C*
      DIMENSION E(NKE),D(NKD),A(NKA),PDED(NKE),PDDE(NKD),PDDA(NKD),
     X    PKED(2),PKDE(2),PKDA(6),PKAD(6),
     X    PAJD(NKA),AEED(NKE),AEDE(NKD),AEDA(NKD),EA(NKE),EK(NKE),
     X    EH(NKE),DA(NKD),DK(NKD),DH(NKD),AA(NKA),AK(NKA),AH(NKA)
      DIMENSION DA1(2),DA2(2),DH1(2),DH2(2),DK1(2),DK2(2)
C*
      WRITE(6,1)
      WRITE(6, 201)
    1 FORMAT(26H -------------------------)
  201 FORMAT(24H START SUBROUTINE AIRAIR)
C*
      CALL ATRTED(E,D,PDED,PDDE, PKED, PKDE,CA,NKE,NKD,AEED,AEDE,AEDA,
     X            EA,EK,EH,DA1,DK1,DH1,IAFED)
      CALL ATRTDA(DA1,A,PDDA, PKDA, PKAD,PAJD,AEDA,CA,NKA,NKD,DA2,DK2,
     X            DH2,AA,AK,AH,IAFDA)
      DO 90 IKD=1,NKD
      DA(IKD)=DA2(IKD)
      DK(IKD)=DK1(IKD)+DK2(IKD)
      DH(IKD)=DH1(IKD)+DH2(IKD)
   90 CONTINUE
C*
      DO 110 I=1,NKE
      PRINT 210, I,E(I)
      PRINT 211, I,EA(I),I,EK(I),I,EH(I)
  110 CONTINUE
      DO 120 I=1,NKD
      PRINT 220, I,D(I)
      PRINT 221, I,DA1(I),I,DK1(I),I,DH1(I)
      PRINT 222, I,DA2(I),I,DK2(I),I,DH2(I)
      PRINT 223, I, DA(I),I, DK(I),I, DH(I)
  120 CONTINUE
      DO 130 I=1,NKA
      PRINT 230, I,A(I)
      PRINT 231, I, AA(I),I, AK(I),I, AH(I)
  130 CONTINUE
      WRITE(6, 299)
      WRITE(6,2)
  210 FORMAT(6H     E(,I1,2H)=,F10.4)
  211 FORMAT(6H    EA(,I1,2H)=,F10.4,6H    EK(,I1,2H)=,F10.4,6H    EH(,I1,
     X2H)=,F10.4)
  220 FORMAT(6H     D(,I1,2H)=,F10.4)
  221 FORMAT(6H   DA1(,I1,2H)=,F10.4,6H   DK1(,I1,2H)=,F10.4,6H   DH1(,I1,
     X2H)=,F10.4)
  222 FORMAT(6H   DA2(,I1,2H)=,F10.4,6H   DK2(,I1,2H)=,F10.4,6H   DH2(,I1,
     X2H)=,F10.4)
  223 FORMAT(6H    DA(,I1,2H)=,F10.4,6H    DK(,I1,2H)=,F10.4,6H    DH(,I1,
     X2H)=,F10.4)
  230 FORMAT(6H     A(,I1,2H)=,F10.4)
  231 FORMAT(6H    AA(,I1,2H)=,F10.4,6H    AK(,I1,2H)=,F10.4,6H    AH(,I1,
     X2H)=,F10.4)
  299 FORMAT(25H END OF SUBROUTINE AIRAIR)
    2 FORMAT(26H -------------------------)
C
C*
      RETURN
      END
```

```
      SUBROUTINE ATRTAB(AA,SS,AS,AN,PARK,PDS,PDN,FSK,PKAN,PKAS,TPS,N1,
     X                   IEQ,SSK,ASK,ANK)
      DIMENSION AA(N1),PDS(N1),PDN(N1),FSK(N1),PKAN(N1),PKAS(N1),TPS(N1)
      GO TO(100,200,300),IEQ
C     COMPUTE ATTRITION ASSUMING SHELTERS ARE ATTACKED ONLY IF NO OPEN
C     AIRCRAFT ARE DETECTED.
  100 ASK=0.
      ANK=0.
      SSK=0.
      IF(SS.EQ.0.0) GO TO 115
      TSSK=1.
      TASK=1.
      DO 110 IAC=1,N1
      TEXP=TPS(IAC)*AA(IAC)
      TEMP=1.0-(1.0-PDS(IAC))**SS
      TEMP1=(1.0-PDN(IAC))**AN
      TEMP3=AMIN1(PKAS(IAC)/SS,PKAS(IAC))
      TEMP2=(1.0-TEMP3*TEMP*TEMP1)**TEXP
      TASK=TASK*TEMP2
      TEMP3=AMIN1(PKAS(IAC)*FSK(IAC)/SS,PKAS(IAC)*FSK(IAC))
      TEMP2=(1.0-TEMP3*TEMP*TEMP1)**TEXP
      TSSK=TSSK*TEMP2
  110 CONTINUE
      ASK=AS*(1.0-TASK)
      SSK=SS*(1.0-TSSK)
  115 DENOM=AMIN1(PARK,AN)
      IF(DENOM.EQ.0.0) GO TO 999
      TANK=1.0
      DO 120 IAC=1,N1
      TEXP=TPS(IAC)*AA(IAC)
      TEMP=1.0-(1.0-PDN(IAC))**AN
      TEMP3=AMIN1(PKAN(IAC)/DENOM,PKAN(IAC))
      TEMP2=(1.0-TEMP3*TEMP)**TEXP
      TANK=TANK*TEMP2
  120 CONTINUE
      ANK=AN*(1.0-TANK)
      GO TO 999
C     COMPUTE ATTRITION ASSUMING OPTIMAL  ALLOCATION OF ATTACKERS TO
C     TARGETS -- COMPUTE ALLOCATION OF ATTACKERS TO SHELTERED AND
C     NON-SHELTERED AIRCRAFT.
  200 ASK=0.
      ANK=0.
      SSK=0.
      DENOM=AMIN1(PARK,AN)
      IF(DENOM.EQ.0.0.AND.SS.EQ.0.0) GO TO 999
      FAAAN=1.
      IF(SS.EQ.0.0) GO TO 225
      FAC=1.
      IF(DENOM .LT. .00005) GO TO 215
      TFAC=1.
      SFAC=1.
      DO 210 IAC=1,N1
      TEXP=TPS(IAC)*AA(IAC)
      TEMP=1.0-(1.0-PDN(IAC))**AN
      TEMP3=AMIN1(PKAN(IAC)/DENOM,PKAN(IAC))
      TEMP2=(1.0-TEMP3*TEMP)**TEXP
      TFAC=TFAC*TEMP2
```

```
      TEMP=1.0-(1.0-PDS(IAC))**SS
      TEMP3=AMIN1(PKAS(IAC)/SS,PKAS(IAC))
      TEMP2=(1.0-TEMP3*TEMP)**TEXP
      SFAC=SFAC*TEMP2
  210 CONTINUE
      TEMP=(AN*TFAC*ALOG(TFAC))/(AS*ALOG(SFAC))
      FAAAN=1.0-ALOG(TEMP)/(ALOG(TFAC)+ALOG(SFAC))
      FAAAN=AMAX1(0.0,FAAAN)
      FAAAN=AMIN1(1.0,FAAAN)
      FAC=1.0-FAAAN
C     COMPUTE ATTRITION
  215 TASK=1.
      TSSK=1.
      DO 220 IAC=1,N1
      TEXP=TPS(IAC)*AA(IAC)*FAC
      TEMP=1.0-(1.0-PDS(IAC))**SS
      TEMP3=AMIN1(PKAS(IAC)/SS,PKAS(IAC))
      TEMP2=(1.0-TEMP3*TEMP)**TEXP
      TASK=TASK*TEMP2
      TEMP3=AMIN1(PKAS(IAC)*FSK(IAC)/SS,PKAS(IAC)*FSK(IAC))
      TEMP2=(1.0-TEMP3*TEMP)**TEXP
      TSSK=TSSK*TEMP2
  220 CONTINUE
      ASK=AS*(1.0-TASK)
      SSK=SS*(1.0-TSSK)
      IF(DENOM .LT. .00005) GO TO 999
  225 TANK=1.0
      DO 230 IAC=1,N1
      TEXP=TPS(IAC)*AA(IAC)*FAAAN
      TEMP=1.0-(1.0-PDN(IAC))**AN
      TEMP3=AMIN1(PKAN(IAC)/DENOM,PKAN(IAC))
      TEMP2=(1.0-TEMP3*TEMP)**TEXP
      TANK=TANK*TEMP2
  230 CONTINUE
      ANK=AN*(1.0-TANK)
      GO TO 999
C     COMPUTE ATTRITION ASSUMING SHELTERS AND OPEN AIRCRAFT ARE ON SAME
C     PARKING AREAS.
  300 ASK=0.
      ANK=0.
      SSK=0.
      TAC=SS+AN
      DENOM=AMIN1(PARK,TAC)
      IF(TAC.EQ.0.0.OR.DENOM.EQ.0.0) GO TO999
      TASK=1.0
      TANK=1.0
      TSSK=1.0
      DO 310 IAC=1,N1
      TEXP=TPS(IAC)*AA(IAC)
      TEMP=PDN(IAC)*AN+PDS(IAC)*SS
      TEMP1=1.0-(1.0-(TEMP/TAC))**TAC
      TEMP3=AMIN1(PKAS(IAC)/DENOM,PKAS(IAC))
      TEMP2=(1.0-TEMP3*TEMP1)**TEXP
      TASK=TASK*TEMP2
      TEMP3=AMIN1(PKAS(IAC)*FSK(IAC)/DENOM,PKAS(IAC)*FSK(IAC))
      TEMP2=(1.0-TEMP3*TEMP1)**TEXP
      TSSK=TSSK*TEMP2
      TEMP3=AMIN1(PKAN(IAC)/DENOM,PKAN(IAC))
      TEMP2=(1.0-TEMP3*TEMP1)**TEXP
      TANK=TANK*TEMP2
  310 CONTINUE
      ASK=AS*(1.0-TASK)
      SSK=SS*(1.0-TSSK)
      ANK=AN*(1.0-TANK)
  999 RETURN
      END
```

```
C* DECK ATRTDA
      SUBROUTINE ATRTDA(D,AT,   PDD,PKD,PKA,    PAJO,
     X       AED,CA,N1,N2,DA,DK,DH,   AA,AK,AH,IATRTF)
C
C        ATRTDA COMPUTES ATTRITION FOR DEFENDERS VERSUS ATTACKERS.
C
C        THE INPUTS ARE:
C             D=NUMBER OF DEFENDERS
C             AT=NUMBER OF ATTACKERS
C             PDD=PROBABILITY THAT A DEFENDER WILL DETECT AN ATTACKER
C             PKD=PROBABILITY THAT A DEFENDER WILL KILL AN
C                 ATTACKER IF ENGAGED
C             PKA=PROBABILITY THAT AN ATTACKER WILL KILL A
C                 DEFENDER IF ENGAGED
C             PAJO=PROBABILITY THAT AN ATTACKER WHEN ENGAGED WILL JETTI-
C                  SON ITS ORDNANCE AND RETURN  FIRE
C             AED=AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION
C                 TO 1.0) THAT A DEFENDER CAN POTENTIALLY MAKE.
C             CA=COMBAT AREAS
C             N1=DIMENSION FOR ATTACKER VARIABLES
C             N2=DIMENSION FOR DEFENDER VARIABLES
C
C        OUTPUT VARIABLES:
C             DA=DEFENDERS ALIVE
C             DK=DEFENDERS KILLED
C             DH=DEFENDERS GONE HOME
C             AA=ATTACKERS ALIVE
C             AK=ATTACKERS KILLED
C             AH=ATTACKERS GONE HOME
C             IATRTF=ATTRITION FUNCTION
C
C        IN THIS ROUTINE THE DEFENDERS SHOOT AT THE ATTACKERS.
C        THE ATTACKERS CAN SHOOT BACK WHEN
C        THEY ARE SHOT AT.
C
      DIMENSION D(N2),AT(N1),PAJO(N1),
     1 AED(N2),PDD(N2),PKD(N2,N1),PKA(N1,N2)
      DIMENSION DA(N2),DK(N2),DH(N2),       AA(N1),AK(N1),AH(N1)
      DIMENSION PKK(6),FK(3),DD(2)
C
C        COMPUTE DD, THE NUMBER OF POSSIBLE SHOTS THE DEFENDERS HAVE
C
      DO 5 KAC=1,N2
      DD(KAC)=D(KAC)*(1.+AED(KAC))
    5 CONTINUE
C
C        SUM THE NUMBER OF AIRCRAFT.  IF THERE ARE LESS THAN .0001 SKIP
C        AROUND THE ROUTINE AND SET OUTPUT VARIABLES TO ZERO.
C
      TT=0.
      DO 10 IAC=1,N1
      TT=TT+AT(IAC)
   10 CONTINUE
      IF(TT.LE..0001)GO TO 70
C
C        SET UP  THE PROBABILITY OF DETECTING AND KILLING AN ATTACKER
C        VARIABLES FOR BINFAC
```

```
C
      DO 25 KAC=1,N2
      DO 24 IAC=1,N1
      IADDR=(IAC-1)*N2+KAC
      PKK(IADDR)=PKD(KAC,IAC)
   24 CONTINUE
   25 CONTINUE
      TEMP=TT/CA
      CALL BINFAC(DD,PDD,PKK,TEMP,CA,N2,N1,FK)
      DO 30 IAC=1,N1
      AK(IAC)=AT(IAC)*(1.-FK(IAC))
   30 CONTINUE
C
C         SET UP THE PROBABILITY OF AN ATTACKER JETTISONING ITS ORDNANCE
C         VARIABLE FOR BINFAC. GET THE FRACTION THAT WILL JETTISON.
C
      DO 40 KAC=1,N2
      DO 40 IAC=1,N1
      IADDR=(IAC-1)*N2+KAC
      PKK(IADDR)=PAJO(IAC)
   40 CONTINUE
      CALL BINFAC(DD,PDD,PKK,TEMP,CA,N2,N1,FK)
      DO 45 IAC=1,N1
C
C         THE NUMBER OF ATTACKERS THAT WILL GO HOME IS THE FRACTION OF
C         TOTAL ATTACKERS THAT JETTISON ORDANCE AND RETURN FIRE.
C
      AH(IAC)=AT(IAC)*(1.0-FK(IAC))
   45 CONTINUE
C
C         IF THE TOTAL SHOTS FIRED (TEMT) BY THE DEFENDERS IS NEGLIGIBLE
C         SET ATTACKER ATTRITION VARIABLES TO ZERO.
C
      TEMT=0.
      DO 57 KAC=1,N2
      TEMT=TEMT+DD(KAC)*PDD(KAC)
   57 CONTINUE
      IF(TEMT.LE..0001) GO TO 70
C
C         THE NUMBER OF TYPE KAC DEFENDERS KILLED AND DAMAGED (DD) IS THE
C         SUM OF THE NUMBER OF TYPE IAC ATTACKERS THAT
C         SHOT AT THE DEFENDER (DK) TIMES THE ALLOCATION OF SHOTS (TEMP1)
C         FOR THOSE ATTACKERS.                 THE ALLOCATION OF SHOTS
C         DEPENDS ON THE NUMBER OF DEFENDERS, PROBABILITY OF DETECTION
C         AND THE POSSIBLE SHOTS THAT COULD BE FIRED.
C
      DO 60 KAC=1,N2
      DK(KAC)=0.
      DO 59 IAC=1,N1
      DK(KAC)=DK(KAC)+AMAX1(0.0,AH(IAC)-AK(IAC))*PKA(IAC,KAC)
      DK(KAC)=AMIN1(DK(KAC),D(KAC))
   59 CONTINUE
C
C         THE PROPORTION OF DEFENDERS THAT DETECT ENEMY AIRCRAFT AND ARE
C                    NOT KILLED GO HOME. DEFENDERS ALIVE ARE THOSE THAT
C                   ARE NOT KILLED OR THAT GO HOME.
C
```

```
      IF(PDD(KAC).LT.1.0) GO TO 61
      DH(KAC)=D(KAC)
      GO TO 62
   61 DH(KAC)=D(KAC)*(1.0-(1.0-PDD(KAC))**TEMP)
   62 DH(KAC)=AMAX1(0.0,DH(KAC)-DK(KAC))
      DA(KAC)=AMAX1(0.0,D(KAC)-DK(KAC)-DH(KAC))
   60 CONTINUE
C
C        THE ATTACKER                  OUTPUT VARIABLES WERE CALCULATED
C        FROM BINFAC. HERE SUBTRACT OUT EMBEDDED VALUES TO GET TRUE
C        VALUES.
C
      DO 65 IAC=1,N1
      AH(IAC)=AMAX1(0.0,AH(IAC)-AK(IAC))
      AA(IAC)=AMAX1(0.0,AT(IAC)-AK(IAC)-AH(IAC))
   65 CONTINUE
      GO TO 99
C
C        ENTER THIS SECTION ONLY IF THERE ARE NOT ENOUGH AIRPLANES TO
C        JUSTIFY DOING ATTRITION.
C
   70 DO 72 IAC=1,N1
      AA(IAC)=AT(IAC)
      AH(IAC)=0.
      AK(IAC)=0.
   72 CONTINUE
      DO 73 KAC=1,N2
      DA(KAC)=D(KAC)
      DK(KAC)=0.
      DH(KAC)=0.
   73 CONTINUE
   99 CONTINUE
      RETURN
      END
```

```
C* DECK ATRTED
      SUBROUTINE ATRTED(ET,DT,PDE,PDD,PKE,PKD,          CA,N1,N2,AEE,
     X AED,AEA,EA,EK,    EH,DA,DK,    DH,IAF)
C
C        ATRTED COMPUTES ATTRITION FOR ESCORTS VERSES DEFENDERS.
C
C        THE INPUTS ARE:
C             ET=TOTAL NUMBER OF ESCORTS
C             DT=TOTAL NUMBER OF DEFENDERS
C             PDE=PROBABILITY THAT AN ESCORT WILL DETECT A DEFENDER
C             PDD=PROBABILITY THAT A DEFENDER WILL DETECT AN ESCORT
C             PKE=PROBABILITY THAT AN ESCORT WILL KILL A DEFENDER
C             PKD=PROBABILITY THAT A DEFENDER WILL KILL AN ESCORT
C             CA=COMBAT AREAS
C             N1=DIMENSION FOR ESCORT VARIABLES
C             N2=DIMENSION FOR DEFENDER VARIABLES
C             AEE=AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION
C                 TO 1.0) AN ESCORT CAN SHOOT AT A DEFENDER
C             AED=AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION
C                 TO 1.0) A DEFENDER CAN SHOOT AT AN ESCORT
C             AEA=AVERAGE NUMBER OF ADDITIONAL ENGAGEMENTS (IN ADDITION
C                 TO 1.0) A DEFENDER CAN SHOOT AT AN ATTACKER
C             IAF=INDEX FOR ATTRITION CALCULATIONS
C
C        THE OUTPUTS ARE:
C             EA=ESCORTS ALIVE
C             EK=ESCORTS KILLED
C             EH=ESCORTS GONE HOME
C             DA=DEFENDERS ALIVE
C             DK=DEFENDERS KILLED
C             DH=DEFENDERS GONE HOME
C
C        IN THIS ROUTINE A SHOOT AND SHOOT BACK SCHEME IS USED.  THE
C        DEFENDERS SHOOT AT THE ESCORTS FIRST, AND THEN ESCORTS LEFT
C        SHOOT BACK.  THEN THE ESCORTS SHOOT AT THE DEFENDERS FIRST AND
C        THE NUMBER OF DEFENDERS LEFT SHOOTS BACK.  THE ATTRITION
C        RESULTS ARE AVERAGED AS EXPLAINED LATER.
C
      DIMENSION EA(N1),DA(N2),PDE(N1),PDD(N2),PKE(N1,N2),PKD(N2,N1),
     X     EK(N1),EH(N1),DK(N2),DH(N2),DK1(2),DK2(2),EK1(1),EK2(1),
     X     AED(N2),AEA(N2),AEE(N1),ET(N1),DT(N2)
      DIMENSION E(1),D(2),AAED(2),BEE(1)
C
C        SUM THE NUMBER OF ESCORTS AND DEFENDERS. IF THERE ARE LESS THAN
C        .0001 OF EITHER SKIP AROUND ROUTINE AND SET ATTRITION VARIABLES
C        TO ZERO.
C
      TEMP1=0.
      DO 2 IAC=1,N1
    2 TEMP1=TEMP1+ET(IAC)
      IF(TEMP1.LE.0.0001) GO TO 90
      TEMP=0.
      DO 4 KAC=1,N2
    4 TEMP=TEMP+DT(KAC)
      IF(TEMP.LE.C.0001) GO TO 90
C
C        THIS SECTION CALCULATES AAED. IT IS A PORTION OF THE TOTAL
```

```
C         ADDITIONAL SHOTS FOR THE DEFENDERS
C
      TEMPA=0.0
      DO 5 KAC=1,N2
      AAED(KAC)=0.0
    5 TEMPA=TEMPA+DT(KAC)*AED(KAC)
      IF(TEMPA.LE.0.0) GO TO 8
      DO 7 KAC=1,N2
      TEMPB=((DT(KAC)*AED(KAC)/TEMPA)*(TEMP1-DT(KAC)))
      IF(DT(KAC).GT.0.0) TEMPB=TEMPB/DT(KAC)
      TEMPB=AMAX1(0.0,TEMPB)
      AAED(KAC)=AMIN1(AED(KAC),TEMPB)
    7 CONTINUE
    8 CONTINUE
C
C          THIS SECTION CALCULATES BEE, WHICH IS A PORTION OF THE TOTAL
C          ADDITIONAL SHOTS FOR THE DEFENDERS
C
      IF(IAF.EQ.0) GO TO 12
C
      TEMPA = 0.0
      DO 9 IAC=1,N1
      BEE(IAC)=0.0
    9 TEMPA=TEMPA+ET(IAC)*AEE(IAC)
      IF(TEMPA.LE.0.0) GO TO 11
      DO 10 IAC=1,N1
      TEMPB=((ET(IAC)*AEE(IAC)/TEMPA)*(TEMP-ET(IAC)))
      IF(ET(IAC).GT.0.0) TEMPB=TEMPB/ET(IAC)
      TEMPB=AMAX1(0.0,TEMPB)
      BEE(IAC)=AMIN1(AEE(IAC),TEMPB)
   10 CONTINUE
   11 CONTINUE
      GO TO 14
C
   12 DO 13 IAC=1,N1
   13 BEE(IAC) = AEE(IAC)
   14 CONTINUE
C
C         LET THE DEFENDERS SHOOT FIRST. SET UP THE INPUTS FOR BINOAT.
C         D IS THE TOTAL POSSIBLE SHOTS FOR THE DEFENDERS.
C
      DO 15 KAC=1,N2
      D(KAC)=DT(KAC)*(1.0+AAED(KAC))
   15 CONTINUE
      CALL BINOAT(D ,ET,PDD,PKD,CA,N2,N1,EK1)
C
C         LET THE NUMBER OF ESCORTS LEFT SHOOT BACK AT THE DEFENDERS
C         WITH ALL OF THEIR ADDITIONAL SHOTS.
C
      DO 25 IAC=1,N1
      E(IAC)=ET(IAC)-EK1(IAC)
      E(IAC)=E(IAC)*(1.0+BEE(IAC))
   25 CONTINUE
      CALL BINOAT(E,DT,PDE,PKE,CA,N1,N2,DK2)
C
C         LET THE ESCORTS SHOOT FIRST WITH ALL ADDITIONAL SHOTS.
C
```

```
      DO 61 IAC=1,N1
   61 E(IAC)=ET(IAC)*(1.0+BEE(IAC))
      CALL BINOAT(E,DT,PDE,PKE,CA,N1,N2,DK1)
C
C        LET THE NUMBER OF DEFENDERS LEFT SHOOT BACK AT THE ESCORTS
C        WITH A PORTION OF THEIR ADDITIONAL SHOTS.
C
      DO 63 KAC=1,N2
      D(KAC)=DT(KAC)-DK1(KAC)
      D(KAC)=D(KAC)*(1.0+AAED(KAC))
   63 CONTINUE
      CALL BINOAT(D,ET,PDD,PKD,CA,N2,N1,EK2)
C
C        ESF AND DSF ARE THE PROPORTION OF ESCORTS THAT SHOOT FIRST AND
C        THE PROPORTION OF DEFENDERS THAT SHOOT FIRST RESPECTIVELY.
C        TEMP AND TEMP1 ARE THE NUMBER OF DEFENDERS PER COMBAT AREA AND
C        THE NUMBER OF ESCORTS PER COMBAT AREA RESPECTIVELY.
C
      ESF=0.
      DSF=0.
      DO 32 IAC=1,N1
   32 ESF=ESF+ET(IAC)*(AEE(IAC)+1.)
      DO 34 KAC=1,N2
   34 DSF=DSF+DT(KAC)*(AED(KAC)+1.)
      EDSUM=ESF+DSF
      ESF=ESF/EDSUM
      DSF=DSF/EDSUM
      TEMP=TEMP/CA
      TEMP1=TEMP1/CA
      DO 40 IAC=1,N1
      EK(IAC)=EK1(IAC)*DSF+EK2(IAC)*ESF
      IF(PDE(IAC).LT.1.0) GO TO 44
      EH(IAC)=ET(IAC)
      GO TO 46
   44 EH(IAC)=ET(IAC)*(1.0-(1.0-PDE(IAC))**TEMP)
   46 EH(IAC)=EH(IAC)*(1.0+BEE(IAC))/(1.0+AEE(IAC))
      EH(IAC)=AMAX1(0.0,EH(IAC)-EK(IAC))
      EA(IAC)=ET(IAC)-EK(IAC)-EH(IAC)
   40 CONTINUE
C
C        DEFENDERS HOME IS THE PERCENTAGE OF LIVE DEFENDERS THAT MAKE A
C        DETECTION TIMES THE FRACTION OF MISSILES ACTUALLY FIRED, (IE.
C        IF 1/3 OF THE MISSILES ARE FIRED 1/3 OF THE DEFENDERS LEFT
C        ALIVE THAT MADE A DETECTION GO HOME.)
C
      DO 50 KAC=1,N2
      DK(KAC)=DK1(KAC)*ESF+DK2(KAC)*DSF
      DH(KAC)=DT(KAC)*(1.0-(1.0-PDD(KAC))**TEMP1)
      DH(KAC)=DH(KAC)*(1.0+AAED(KAC))/(1.0+AEA(KAC))
      DH(KAC)=AMAX1(0.0,DH(KAC)-DK(KAC))
      DA(KAC)=DT(KAC)-DH(KAC)-DK(KAC)
   50 CONTINUE
      GO TO 99
C
C        ENTER THIS SECTION ONLY IF THERE ARE A NEGLIGIBLE NUMBER OF
C        ESCORTS OR DEFENDERS
C
```

SUBROUTINE ATRTED

```
   90 DO 92 IAC=1,N1
      EA(IAC)=ET(IAC)
      EH(IAC)=0.
   92 EK(IAC)=0.
      DO 94 KAC=1,N2
      DA(KAC)=DT(KAC)
      DH(KAC)=0.
   94 DK(KAC)=0.
   99 CONTINUE
      RETURN
      END
```

SUBROUTINE ATRTIA

```
C* DECK ATRTIA
      SUBROUTINE ATRTIA(DLI,CAP,ESC,BMR,DLIM,CAPM,ESCM,PKFA,PKEF,FK,EK,
     1                  BK,IAF)
C*
      WRITE(6,1)
      WRITE(6, 201)
    1 FORMAT(26H --------------------------)
  201 FORMAT(24H START SUBROUTINE ATRTIA)
C*
      FTR = CAP+DLI
      ATT = ESC + BMR
      IF(FTR.GT.0..AND.ATT.GT.0.) GO TO 10
      FK = 0.
      BK = 0.
      EK = 0.
      GO TO 100
   10 CONTINUE
C*
      FT1 = AMAX1(FTR,1.0)
      AT1 = AMAX1(ATT,1.0)
C*
C*    SELECT ATTRITION EQUATION
C*    IAF=1 MEANS RANDOM TARGETING,  IAF=2 MEANS COORDINATED TARGETING
C*
      IF(IAF.EQ.2) GO TO 50
C*
C*    CALCULATIONS FOR IAF = 1 --
C*
C*    LET FIGHTERS SHOOT FIRST
C*
      FM = CAP*CAPM + DLI*DLIM
      BK1 = BMR*(1.-(1.-PKFA/AT1)**FM)
      EK1 = ESC*(1.-(1.-PKFA/AT1)**FM)
      EM1 = (ESC-EK1)*ESCM
      FK1 = FTR*(1.-(1.-PKEF/FT1)**EM1)
C*
C*    LET ESCORTS SHOOT FIRST
C*
      EM = ESC*ESCM
      DK2 = DLI*(1.-(1.-PKEF/FT1)**EM)
      CK2 = CAP*(1.-(1.-PKEF/FT1)**EM)
      FM2 = (CAP-CK2)*CAPM + (DLI-DK2)*DLIM
      BK2 = BMR*(1.-(1.-PKFA/AT1)**FM2)
      EK2 = ESC*(1.-(1.-PKFA/AT1)**FM2)
C*
      GO TO 90
C*
   50 CONTINUE
C*
C*    CALCULATIONS FOR IAF = 2 --
C*
C*    LET FIGHTERS SHOOT FIRST
C*
      FM =  CAP*CAPM + DLI*DLIM
      SDT = FM/AT1
      TSDT = AINT(SDT)
      BK1 = BMR*(1.-((1.-(SDT-TSDT)*PKFA)*((1.-PKFA)**TSDT)))
```

```
      EK1 = ESC*(1.-((1.-(SDT-TSDT)*PKFA)*((1.-PKFA)**TSDT)))
      EM1 = (ESC-EK1)*ESCM
      SDT = EM1/FT1
      TSDT = AINT(SDT)
      FK1 = FTR*(1.-((1.-(SDT-TSDT)*PKEF)*((1.-PKEF)**TSDT)))
C*
C*    LET ESCORTS SHOOT FIRST
C*
      EM = ESC*ESCM
      SDT = EM/FT1
      TSDT = AINT(SDT)
      DK2 = DLI*(1.-((1.-(SDT-TSDT)*PKEF)*((1.-PKEF)**TSDT)))
      CK2 = CAP*(1.-((1.-(SDT-TSDT)*PKEF)*((1.-PKEF)**TSDT)))
      FM2 = (CAP-CK2)*CAPM + (DLI-DK2)*DLIM
      SDT = FM2/AT1
      TSDT = AINT(SDT)
      BK2 = BMR*(1.-((1.-(SDT-TSDT)*PKFA)*((1.-PKFA)**TSDT)))
      EK2 = ESC*(1.-((1.-(SDT-TSDT)*PKFA)*((1.-PKFA)**TSDT)))
C*
   90 CONTINUE
C*
C*    AVERAGE THE RESULTS
C*
      EK = (EK1+EK2)/2.
      BK = (BK1+BK2)/2.
      FK = (FK1+DK2+CK2)/2.
C*
  100 CONTINUE
C*
      PRINT 210,FK,EK,BK
  210 FORMAT(8H     FK=,F10.4,8H     EK=,F10.4,8H     BK=,F10.4)
      WRITE(6, 299)
      WRITE(6,2)
  299 FORMAT(25H END OF SUBROUTINE ATRTIA)
    2 FORMAT(26H -------------------------)
C
      RETURN
      END
```

```
C* DECK ATRTSS
      SUBROUTINE ATRTSS(T,AVGSS,A,PDA,PSA,PKA,AVLS,ANM,PDS,PKS,FASS,CA,
     1NX,N1,AESGS,FSM,FVS,TSC,IAF,IAW,SA,SS,SK,AA,AH,AK)
C
C         ATRTSS COMPUTES ATTRITION FOR AIRCRAFT VERSUS SAMS
C
C         THE INPUTS ARE:
C              T=TOTAL NUMBER OF SAMS
C              AVGSS=AVERAGE NUMBER OF SAM SHOTS PER SAM PER AIRCRAFT
C              A=NUMBER OF AIRCRAFT
C              PDA=PROBABILITY THAT A SAM SUPPRESSOR WILL DETECT A SAM
C              PSA=PROBABILITY THAT A SAM SUPPRESSOR WILL SUPPRESS A SAM
C              PKA=PROBABILITY THAT A SAM SUPPRESSOR WILL KILL A SAM
C              AVLS=AVAILABILITY FACTOR FOR SAMS
C              ANM=ACTUAL NUMBER OF MISSLES FOR SAMS
C              PDS=PROBABILITY THAT A SAM WILL DETECT AN AIRCRAFT
C              PKS=PROBABILITY THAT A SAM WILL KILL AN AIRCRAFT
C              FASS=FRACTION OF AIRCRAFT THAT CAN DO SAM SUPPRESSION
C              CA=COMBAT AREAS
C              NX=DIMENSION OF SAM VARIABLES (ILS)
C              N1=DIMENSION OF AIRCRAFT VARIABLES (IAC)
C              AESGS=AVERAGE NUMBER OF ADDITIONAL GROUND TARGETS THAT AN
C                    AIRCRAFT CAN POTENTIALLY MAKE AFTER HAVING
C                    ENGAGED A SAM
C              FSM=FRACTION OF AIRCRAFT THAT ARE INVULNERABLE TO SAMS
C                  (PERHAPS DUE TO STANDOFF MUNITIONS)
C              FVS=FRACTION OF SAMS VULNERABLE TO SAM SUPPRESSORS
C              TSC=TOTAL SHOTS PER CYCLE FOR A SAM
C              IAF=ATTRITION FUNCTION.(VARIABLE IS NOT CURRENTLY USED.)
C              IAW=ATTRITION WEIGHTING METHOD
C
C         THE OUTPUT VARIABLES ARE:
C              SA=SAMS ALIVE
C              SS=SAMS SUPPRESSED
C              SK=SAMS KILLED
C              AA=AIRCRAFT ALIVE
C              AH=AIRCRAFT GONE HOME
C              AK=AIRCRAFT KILLED
C
C         THIS ROUTINE USES A SHOOT AND SHOOT BACK SCHEME. FIRST ALL THE
C         AIRCRAFT SHOOT AT THE SAMS THAT ARE VULNERABLE. THEN THE SAMS
C         LEFT SHOOT BACK AT THE AIRCRAFT WITHOUT STANDOFF MUNITIONS.
C         NEXT THE SAMS ARE ALLOWED TO SHOOT FIRST, AND THE AIRCRAFT
C         SHOOT BACK.
C
      DIMENSION T(NX),AVGSS(NX),A(N1),PDA(N1),PSA(N1,NX),PKA(NX),
     X    AVLS(NX),ANM(NX),PDS(NX),PKS(NX,N1),FASS(N1),SA(NX),SS(NX),
     X    SK(NX),        AA(N1),AH(N1),AK(N1),        AESGS(N1),
     X    SK1(2),SK2(2),                SL(2),SS1(2),SS2(2),
     X    AK1(3) ,AK2(3) ,                  AL(3) ,ST(3) ,AT(3) ,
     X    FSM(N1),FVS(NX),TSC(NX)
      DIMENSION  PK(6),S(2),   SF1(2),SF2(2),ANM1(2),ANM2(2)
C*
      WRITE(6,1)
      WRITE(6, 201)
    1 FORMAT(26H -------------------------)
  201 FORMAT(24H START SUBROUTINE ATRTSS)
```

```
C*
C*        CONSIDER ONLY AVAILABLE SAMS
C*
      DO 3 ILS=1,NX
    3 S(ILS) = T(ILS)*AVLS(ILS)
C
C         SUM THE NUMBER OF SAMS AND AIRCRAFT. IF THERE ARE FEWER THAN
C         .0001 SKIP THE WHOLE ROUTINE, AND SET THE ARRAYS TO ZERO.
C
      ASUM=0.0
      SSUM=0.0
      DO 65 IAC=1,N1
   65 ASUM=ASUM+A(IAC)
      DO 66 ILS=1,NX
      IF(IAW .EQ. 2) GO TO 67
      SSUM=SSUM+S(ILS)*TSC(ILS)
      GO TO 66
   67 SSUM=SSUM+S(ILS)
   66 CONTINUE
      IF(ASUM.LE..0001.OR.SSUM.LE..0001) GO TO 50
C
C         LET ALL OF THE AIRCRAFT SHOOT FIRST AT THE SAMS THAT ARE
C         VULNERABLE TO SAM SUPPRESSORS.
C
C         CONSIDER VULNERABILITY OF SAMS AND SUPPRESSION CAPABILITY OF
C                                                              AIRCRAFT
C
      DO 5 ILS=1,NX
    5 ST(ILS)=S(ILS)*FVS(ILS)
      DO 6 IAC=1,N1
    6 AL(IAC) = A(IAC)*FASS(IAC)
C
      CALL BINOAT(AL,ST,PDA,PSA,CA,N1,NX,SS1)
C
C
C         SUM THE NUMBER OF AIRCRAFT WITHOUT STANDOFF MUNITIONS. IF THERE
C         ARE NONE DO NOT ALLOW THE SAMS TO SHOOT BACK AT THE AIRCRAFT,
C         BUT LET THE TOTAL NUMBER OF SAMS         KILLED AND SUPPRESSED
C               BE SK1 AND SS1 RESPECTIVELY, AND SET THE NUMBER OF
C               AIRCRAFT KILLED TO ZERO.
C
C
      ATSUM=0.0
      DO 16 IAC=1,N1
      AT(IAC)=A(IAC)*(1.0-FSM(IAC))
      ATSUM=ATSUM+AT(IAC)
   16 CONTINUE
      IF(ATSUM.EQ.0.0) GO TO 22
C
C         ONLY SAMS THAT HAVE BEEN SUPPRESSED CAN POTENTIALLY BE KILLED
C                     SO SAMS KILLED  IS CALCULATED BY APPLYING THE
C         PROBABILITY OF KILL TO THE
C         SUPPRESSED SAMS (WHICH WAS CALCULATED IN BINOAT). SAMS LEFT
C         IS THE TOTAL NUMBER OF SAMS
C         MINUS SAMS SUPPRESSED.  THE
C         SHOTS THAT CAN BE FIRED BY THE SAMS LEFT IS AT LEAST ONE, OR
C         PROPORTIONAL TO THE SIZE OF THE RAID UP TO AS MANY MISSILES AS
```

```
C          THEY HAVE LEFT.
C                      THE ACTUAL NUMBER OF MISSLES IS REDUCED BY THE SHOTS
C          FIRED AND THE MISSLES AT SAM SITES THAT WERE KILLED. IF THE
C          TOTAL SHOTS PER CYCLE GOES TO ZERO ALL THE SAMS ARE SUPPRESSED.
C          ADD TO THE SAMS SUPPRESSED
C          THE PROPORTION OF SAMS THAT HAVE RUN OUT OF MISSLES.
C
      DO 11 ILS=1,NX
      SK1(ILS)=SS1(ILS)*PKA(ILS)
      SL(ILS)=S(ILS)-AMAX1(SK1(ILS),SS1(ILS))
      SF1(ILS)=AMAX1(SL(ILS),SL(ILS)*AVGSS(ILS)*ATSUM)
      SF1(ILS)=AMIN1(SF1(ILS),ANM(ILS),SL(ILS)*TSC(ILS))
      TEMP=SF1(ILS)*(1.0-(1.0-PDS(ILS))**(ATSUM/CA))
      ANM1(ILS)=AMAX1(0.0,ANM(ILS)-TEMP-SK1(ILS)*TSC(ILS))
      IF(TSC(ILS).LE.0.0) GO TO 36
      SS1(ILS)=SS1(ILS)+AMIN1(TEMP/TSC(ILS),SL(ILS))
      GO TO 37
   36 SS1(ILS)=SL(ILS)+SS1(ILS)
   37 CONTINUE
   11 CONTINUE
      DO 12 ILS=1,NX
      DO 12 IAC=1,N1
      IADDR=(IAC-1)*NX+ILS
      PK(IADDR)=PKS(ILS,IAC)
   12 CONTINUE
C
C          LET THE SAMS LEFT SHOOT SF1 NUMBER OF MISSLES AT THE PLANES
C          WITHOUT STANDOFF MUNITIONS.
C
      CALL BINOAT(SF1,AT,PDS,PK,CA,NX,N1,AK2)
C
C          LET THE SAMS SHOOT FIRST
C          A SAM CAN SHOOT AT LEAST 1 SHOT, OR IT CAN SHOOT PROPORTIONAL
C          TO THE NUMBER OF AIRCRAFT UP TO AS MANY MISSLES AS IT HAS.
C
      DO 17 ILS=1,NX
      SF2(ILS)=AMAX1(S(ILS),S(ILS)*AVGSS(ILS)*ATSUM)
      SF2(ILS)=AMIN1(SF2(ILS),ANM(ILS),S(ILS)*TSC(ILS))
   17 CONTINUE
      CALL BINOAT(SF2,AT,PDS,PK ,CA,NX,N1,AK1)
C
C          SET THE NUMBER OF AIRCRAFT LEFT TO THE TOTAL NUMBER OF PLANES
C          MINUS THOSE KILLED, AND CONSIDER SUPPRESION CAPABILITY.
C
      DO 14 IAC=1,N1
   14 AL(IAC) = (A(IAC)-AK1(IAC))*FASS(IAC)
C
C          LET THE NUMBER OF AIRCRAFT LEFT SHOOT BACK AT SAMS.
C
      DO 62 ILS=1,NX
      TEMP=SF2(ILS)*(1.0-(1.0-PDS(ILS))**(ATSUM/CA))
      ANM2(ILS)=AMAX1(0.0,ANM(ILS)-TEMP)
      IF(TSC(ILS).LE.0.0) GO TO 63
      SS(ILS)=AMIN1(TEMP/TSC(ILS),S(ILS))
      GO TO 64
   63 SS(ILS)=S(ILS)
   64 CONTINUE
```

```
      ST(ILS)=AMAX1(0.0,ST(ILS)-SS(ILS))
   62 CONTINUE
      CALL BINOAT(AL,ST,PDA,PSA,CA,N1,NX,SS2)
C
C        ASF AND SSF ARE THE PROPORTION OF AIRCRAFT THAT SHOT FIRST, AND
C        THE PROPORTION OF SAMS THAT SHOT FIRST RESPECTIVELY.
C
      ASSUM=ASUM+SSUM
      ASF=ASUM/ASSUM
      SSF=SSUM/ASSUM
C
C                            AIRCRAFT KILLED ARE COMPUTED BY AVERAGING
C        THE NUMBER OF PLANES KILLED WHEN THE SAMS SHOT
C        FIRST AND WHEN THE PLANES SHOT FIRST.
C
      DO 18 IAC=1,N1
      AK(IAC)=AK1(IAC)*SSF+AK2(IAC)*ASF
   18 CONTINUE
C
C        AGAIN COMPUTE THE SAMS KILLED             FROM THOSE THAT HAVE
C        BEEN SUPPRESSED, DECREASE THE ACTUAL NUMBER OF MISSLES AND ADD
C        IN ADDITIONAL SUPPRESSED SAMS
C
      DO 19 ILS=1,NX
      SK2(ILS)=SS2(ILS)*PKA(ILS)
C
C        AS WITH THE AIRPLANES, THE SAMS          KILLED, AND SUPPRESSED
C        ARE AN AVERAGE OF THE SAMS          KILLED, AND SUPPRESSED WHEN
C        THE AIRPLANES SHOT FIRST AND WHEN THE SAMS SHOT FIRST.
C
      SS2(ILS)=SS2(ILS)+SS(ILS)
      SK(ILS)=SK1(ILS)*ASF+SK2(ILS)*SSF
      SS(ILS)=SS1(ILS)*ASF+SS2(ILS)*SSF
      ANM2(ILS)=AMAX1(0.0,ANM2(ILS)-SK2(ILS)*TSC(ILS))
      ANM(ILS)=ANM1(ILS)*ASF+ANM2(ILS)*SSF
   19 CONTINUE
      GO TO 23
C
C        THIS SECTION IS ENTERED ONLY IF ALL THE PLANES USE STANDOFF
C        MUNITIONS
C
   22 CONTINUE
      DO 24 IAC=1,N1
      AK(IAC)=0.0
   24 CONTINUE
      DO 26 ILS=1,NX
      SK(ILS)=SS1(ILS)*PKA(ILS)
      SS(ILS)=SS1(ILS)
   26 CONTINUE
C
C        UP TO THIS POINT SAMS SUPPRESSED INCLUDES SAMS
C        KILLED.  THEY ARE SEPARATED HERE:  SAMS
C        KILLED IS ALREADY A TRUE NUMBER.
C
   23 CONTINUE
      DO 20 ILS=1,NX
      SS(ILS)=AMAX1(0.0,SS(ILS)-SK(ILS))
```

```
      SA(ILS)=AMAX1(0.0,S(ILS)-SS(ILS)-SK(ILS))
      SA(ILS) = SA(ILS) + T(ILS)*(1.-AVLS(ILS))
   20 CONTINUE
C
C         SUM THE NUMBER OF VULNERABLE SAMS. IF THERE ARE NONE THEN NONE
C         OF THE SUPPRESSORS GO HOME BECAUSE THEY ARE OUT OF AMMUNITION.
C         IF THERE ARE SOME VULNERABLE SAMS THEN THE AIRCRAFT THAT GOT
C         ENGAGED AND USED ALL THEIR AMMUNITION GO HOME. THE FRACTION OF
C         PLANES THAT GO HOME EQUALS THE NUMBER OF ENGAGEMENTS DIVIDED BY
C         THE POTENTIAL CAPABILITY.
C
      TOT=0.
      DO 30 ILS=1,NX
      TOT=TOT+ST(ILS)
   30 CONTINUE
      DO 35 IAC=1,N1
      IF(TOT.GT.0.0) GO TO 32
      AH(IAC)=0.0
      GO TO 33
   32 CONTINUE
      TEMP=(1.0-PDA(IAC))**(TOT/CA)
      AH(IAC) = (A(IAC)*FASS(IAC))*(1.0-TEMP)/(1.0+AESGS(IAC))
      AH(IAC)=AMAX1(0.0,AH(IAC)-AK(IAC))
   33 AA(IAC)=AMAX1(0.0,A(IAC)-AH(IAC)-AK(IAC))
   35 CONTINUE
      GO TO 99
C
C         ONLY ENTER THIS SECTION WHEN THERE ARE NO MORE SAMS OR PLANES.
C
   50 CONTINUE
      DO 51 IAC=1,N1
      AA(IAC)=A(IAC)
      AH(IAC)=0.0
      AH(IAC)=0.0
      AK(IAC)=0.0
   51 CONTINUE
      DO 52 ILS=1,NX
      SA(ILS)=S(ILS)
      SS(ILS)=0.0
      SK(ILS)=0.0
   52 CONTINUE
   99 CONTINUE
C*
      DO 110 I=1,NX
      PRINT 210,I,T(I),I,ANM(I)
      PRINT 211,I,SA(I),I,SK(I),I,SS(I)
  110 CONTINUE
      DO 120 I=1,N1
      PRINT 220,I,A(I)
      PRINT 221,I,AA(I),I,AK(I),I,AH(I)
  120 CONTINUE
      WRITE(6, 299)
      WRITE(6,2)
  210 FORMAT(6H    T(,I1,2H)=,F10.4,6H  ANM(,I1,2H)=,F10.4)
  211 FORMAT(6H   SA(,I1,2H)=,F10.4,6H   SK(,I1,2H)=,F10.4,6H   SS(,I1,
     X2H)=,F10.4)
  220 FORMAT(6H    A(,I1,2H)=,F10.4)
  221 FORMAT(6H   AA(,I1,2H)=,F10.4,6H   AK(,I1,2H)=,F10.4,6H   AH(,I1,
     X2H)=,F10.4)
  299 FORMAT(25H END OF SUBROUTINE ATRTSS)
    2 FORMAT(26H -------------------------)
C
C*
      RETURN
      END
```

SUBROUTINE BARKCK

```
      SUBROUTINE BARKCK(SIB,BARLTH,NKP,PEN,EDW,CPK,ECDW,CPCK,ATTWGT,
     1PENK,SIBCK)
      COMMON /BARSCK/  SIBCK1,SIBCK2
      DIMENSION PEN(NKP),EDW(NKP),CPK(NKP),ECDW(NKP),CPCK(NKP),PENK(NKP)
      DIMENSION PENKA(10,2)
C
C   SUBROUTINE BARKCK ASSESSES KILLS BY BARRIER SUBMARINES AGAINST ENEMY
C   PENETRATORS, AND ALSO ASSESSES COUNTERKILLS.
C
C   ITERATION 1--BARRIER SUBMARINES ATTACK FIRST
C   ITERATION 2--PENETRATORS ATTACK FIRST
C   ATTRITION RESULTS REPORTED ARE A WEIGHTED AVERAGE OF THE TWO
C   ITERATIONS
C
      IF(NKP .EQ. 0) GO TO 99
      ITER=1
      SIBA=SIB
    5 DO 20 KP=1,NKP
      CPKKP=CPK(KP)
      SCR=SIBA*AMIN1(BARLTH,EDW(KP))/BARLTH
      NSCR=SCR
      XSCR=NSCR
      FSCR=SCR-XSCR
      IF(NSCR .GT. 0) GO TO 10
      PENKA(KP,ITER)=PEN(KP)*FSCR*CPKKP
      GO TO 20
   10 CONTINUE
      TERM1=(1.-CPKKP)**NSCR
      TERM2=1.-TERM1+TERM1*FSCR*CPKKP
      PENKA(KP,ITER)=PEN(KP)*TERM2
   20 CONTINUE
      IF(ITER .EQ. 2) GO TO 40
C
C   COUNTERKILLS
C
      PROD1=1.
      PROD2=1.
      DO 30 KP=1,NKP
      TERM=AMIN1(1.,ECDW(KP)/BARLTH)
      TERM=TERM*CPCK(KP)
      PROD1=PROD1*(1.-TERM)**(PEN(KP)-PENKA(KP,1))
      PROD2=PROD2*(1.-TERM)**PEN(KP)
   30 CONTINUE
      SIBCK1=SIB*(1.-PROD1)
      SIBCK2=SIB*(1.-PROD2)
      ITER=2
      SIBA=SIB-SIBCK2
      GO TO 5
   40 CONTINUE
      SIBCK=ATTWGT*SIBCK1+(1.-ATTWGT)*SIBCK2
      DO 50 KP=1,NKP
      PENK(KP)=ATTWGT*PENKA(KP,1)+(1.-ATTWGT)*PENKA(KP,2)
   50 CONTINUE
      RETURN
   99 STOP 6401
      END
```

SUBROUTINE BINFAC

```
C* DECK BINFAC
      SUBROUTINE BINFAC(S,D,P,TTC,CA,NI,NJ,FK)
C
C          BINFAC IS A BINOMIAL ATTRITION ROUTINE WHICH COMPUTES A
C          FRACTION NOT KILLED. THE EQUATION USED IS:
C
C                  NI                 TTC                         S(I)/CA
C          FK(J)=MULT(1-(1-(1-D(I))    )P(I,J)/MAX(1,TTC))
C                 I=1
C
C          WHERE THE VARIABLES ARE:
C                FK=FRACTION NOT KILLED
C                S=SHOOTERS
C                D=PROBABILITY OF DETECTION
C                P=PROBABILITY OF KILL
C                TTC=TOTAL TARGETS
C                CA=COMBAT AREAS
C                NI=DIMENSION FOR SHOOTER VARIABLES
C                NJ=DIMENSION FOR TARGET  VARIABLES
C
      DIMENSION S(NI),D(NI),P(NI,NJ),FK(NJ)
C
      DO 20 J=1,NJ
      FK(J)=1.
      IF(TTC.LE.0.0) GO TO 20
      DO 10 I=1,NI
      IF(D(I).LT.1.0) GO TO 4
      TEMP1=0.
      GO TO 6
    4 CONTINUE
      TEMP1=(1.0-D(I))**TTC
    6 CONTINUE
      TEMP2=(1.0-TEMP1)*AMIN1(P(I,J),P(I,J)/TTC)
      IF (S(I).LE.0.0) GO TO 10
      TEMP3=(1.0-TEMP2)**(S(I)/CA)
      FK(J)=FK(J)*TEMP3
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
C* DECK BINOAT
      SUBROUTINE BINOAT(S,T,D,P,C,NSE,NTE,TK)
C
C          BINOAT IS A BINOMIAL ATTRITION ROUTINE. THE EQUATION USED IS:
C
C                  NSE                                     TTC  S(J)/C
C  TK(I)=T(I)*(1-MULT(1-P(J,I)/MAX(1,TTC)*(1-(1-D(J))    ))       )
C                  J=1
C
C          WHERE:
C                 S=NUMBER OF SHOOTERS
C                 T=NUMBER OF TARGETS
C                 D=PROBABILITY OF DETECTION
C                 P=PROBABILITY OF KILL
C                 C=COMBAT AREAS
C                 NSE=DIMENSION OF SHOOTER VARIABLES
C                 NTE=DIMENSION OF TARGET VARIABLES
C                 J=ISE  VARIES FROM 1 TO NSE
C                 I=ITE  VARIES FROM 1 TO NTE
C
C                   NSE
C                 TTC=SUM(T(I)/C)
C                   I=1
C
C                 TK=NUMBER OF TARGETS KILLED
C
      DIMENSION S(NSE),T(NTE),D(NSE),P(NSE,NTE),TK(NTE)
C
      TT=0.
      DO 10 ITE=1,NTE
      TT=TT+T(ITE)
   10 CONTINUE
      IF(TT.EQ.0.) GO TO 25
      TTC=TT/C
      IF(TTC.LT..00005) GO TO 25
      DO 20 ITE=1,NTE
      TOT=1.
      DO 15 ISE=1,NSE
      IF(S(ISE).LE.0.0) GO TO 15
      IF(D(ISE).LT.1.) GO TO 12
      TEMP=0.
      GO TO 13
   12 CONTINUE
      TEMP=(1.0-D(ISE))**TTC
   13 CONTINUE
      IF (S(ISE).LE.0.0) GO TO 15
      TEMP1=AMIN1(P(ISE,ITE),P(ISE,ITE)/TTC)
      TEMP=(1.0-TEMP1*(1.0-TEMP))**(S(ISE)/C)
      TOT=TOT*TEMP
   15 CONTINUE
      TK(ITE)=T(ITE)*(1.0-TOT)
   20 CONTINUE
      RETURN
C
   25 DO 30 ITE=1,NTE
      TK(ITE)=0.
   30 CONTINUE
      RETURN
      END
```

FUNCTION BINOM

```
      FUNCTION BINOM(N,M,P)
C     THIS FUNCTION COMPUTES THE PROBABILITY OF M SUCCESSES IN N
C     TRIALS, WHEN THE PROBABILITY OF SUCCESS ON A TRIAL IS P
      Q=1.-P
      A=P*Q
      IF(A .LT. 0.) GO TO 30
      IF(M-N) 3,1,30
    1 IF(N .NE. 0) GO TO 2
      BINOM=1
      RETURN
    2 CONTINUE
      BINOM=P**N
      RETURN
    3 IF(M .NE. 0) GO TO 4
      BINOM=Q**N
      RETURN
    4 CONTINUE
      IF(A .EQ. 0.) GO TO 25
      X=1.
      IDIF=2*M-N
      IF(IDIF)10,10,20
   10 IDIF=0-IDIF
      DO 12 K=1,M
      KM1=K-1
      X=X*(N-KM1)*A/(M-KM1)
   12 CONTINUE
      BINOM=X*(Q**IDIF)
      RETURN
   20 NMM=N-M
      DO 22 K=1,NMM
      KM1=K-1
      X=X*(N-KM1)*A/(NMM-KM1)
   22 CONTINUE
      BINOM=X*(P**IDIF)
      RETURN
   25 BINOM=0.
      RETURN
   30 STOP 6406
      END
```

SUBROUTINE CTFMOD

```
C* DECK CTFMOD
      SUBROUTINE CTFMOD(L)
C*
C*    CTFMOD EXERCISES THE CTF MODEL BASED ON IDA REPORT R-245
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJD(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10).DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJD(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBP3KS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACP(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XTA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE CTFMOD

```
C*
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C
C* COMDECK COMGA
      COMMON/COMGA/ NTPSLA,BMR(2,3),ESC(2)
C*
C*
C* COMDECK COMSOR
      COMMON/COMSOR/ FTSORU,ATSORU
C*
C*
      DIMENSION AT(3),AT1(3),ATKT(3),FMRBT(4)
      DIMENSION TAB10(20),TAB13(20)
C*
C --- GENERAL FORMATS
      WRITE(6,1)
      WRITE(6,5001)
    1 FORMAT(51H0--------------------------------------------------)
 5001 FORMAT(24H START SUBROUTINE CTFMOD)
C*
C
   10 FORMAT(8F10.2)
   12 FORMAT(8F10.4)
C
C --- FORMATS FOR DISPLAYING PARAMETERS
C
  120 FORMAT(20H0 UBAEWL    UBAEW    )
  130 FORMAT(20H0 AEWD      STAR     )
  140 FORMAT(50H0 CAPML     CAPM      BUCAP     DLIA      WVSIZ    )
  160 FORMAT(40H0 T1        T2        T3        T4       )
  170 FORMAT(40H0 VCAP      CAPMR     TCAP      CAPSTAR )
  200 FORMAT(10H0 BARL    )
  210 FORMAT(20H0 UBASWL    BAREAL   )
  220 FORMAT(20H0 UBASW     BAREA    )
  230 FORMAT(60H0 ASWF      PKASW     ST        PDIN       PKIN      ZLAM
     XPF    )
  240 FORMAT(40H0 PKIIN     ESR       ESLR      SUBSOR   )
  250 FORMAT(60H0 PKSS      FPPL1     PKPL1     PKPL2     PKPLD     FPPL
     X2    )
  260 FORMAT(10H0 TPS     )
  290 FORMAT(20H0 ZMPCAP    ZMPDLI   )
  300 FORMAT(20H0 STG       ZMPSTG   )
  305 FORMAT(10H0 STSALV  )
C*
      ATT=0.
      DO 30 K=1,NKRB
      AT(K) = BMR(1,K) + BMR(2,K)
      ATT = ATT + AT(K)
   30 CONTINUE
      PRINT 32, ATT
   32 FORMAT( 5H ATT=,F10.4)
      IF(ATT.LE.0. .AND. IRSUBA(L).EQ.2) GO TO 2000
      AESC = ESC(1) + ESC(2)
C*
      BAREAL=BARELQ(L)
```

```
      BAREA =BAREAQ(L)
      BARL  =BARLQ(L)
      ESR   =ESRQ(L)
      STAR  =STARQ(L)
      CAPM  =CAPMQ(L)
      CAPML =CAPMLQ(L)
      CAPST =CAPSTQ(L)
      XASWL =XASWLQ(L)
      XAEWL =XAEWLQ(L)
      THSECA=THSCAQ(L)
      THSECT=THSCTQ(L)
      WRRLND=WRLNDQ(L)
      PRWLND=PRWLNQ(L)
C*
      ST   = RS(1,L)*FSTAQ(L)
      STG  = RS(2,L)*FSTGAQ(L)
      IF(ATT.LE.0. .AND. IRSUBA(L).EQ.1) STG = 0.
C
C ---           PRINT PARAMETERS
C
      PRINT 120
      PRINT 12, UBAEWL,UBAEW
C
      PRINT 130
      PRINT10, AEWD,STAR
C
      PRINT 140
      PRINT10,CAPML,CAPM,BUCAP,DLIA,WVSIZ
C
      PRINT 160
      PRINT 10, T1,T2,T3,T4
C
      PRINT 170
      PRINT 10, VCAP,CAPMR,TCAP,CAPST
C
      PRINT200
      PRINT 10, BARL
C
      PRINT 210
      PRINT12,UBASWL,BAREAL
C
      PRINT 220
      PRINT12,UBASW,BAREA
C
      PRINT 230
      PRINT12, ASWF, PKASW, ST, PDIN, PKIN,ZLAMPF
C
      PRINT 240
      PRINT 10, PKIIN, ESR, ESLR, SUBSOR
C
      PRINT 260
      PRINT10,TPS
C
      PRINT 290
      PRINT 10, ZMPCAP,ZMPDLI
C
      PRINT 300
```

```
      PRINT10,STG,ZMPSTG
C
      PRINT 305
      PRINT10,STSALV
C
C --- FORMATS FOR DISPLAYING PARAMETERS OF TABLES
C
  310 FORMAT( 7H0 TAB10)
  330 FORMAT( 7H0 TAB13)
C
C --- FORMATS FOR DISPLAYING SELECTED BLUE RESOURCES
C
  410 FORMAT(40H0 XASW      XAEW      XASWL     XAEWL    )
  420 FORMAT(30H0 XEASWA    XEASWN    XEAAW    )
  430 FORMAT(10H0 XPLAT    )
C
      PRINT 410
      PRINT10,XASW,XAEW,XASWL,XAEWL
C
      PRINT 420
      PRINT10,XEASWA,XEASWN,XEAAW
C
      PRINT 430
      PRINT10,XPLAT
C
C --- FORMATS FOR DISPLAYING COMPUTED VARIABLES
C
  510 FORMAT(14H     XAEWSTA   F10.2)
  520 FORMAT(14H     XCAPSTA   F10.2)
  530 FORMAT(14H     XZ        F10.2)
  540 FORMAT( 7H     XTB,7X,F10.2,71H  XTB CAN BE NEGATIVE.  IF SO, FUNCT
     11 WILL SET APPROPRIATE VALUES TO 0.)
  550 FORMAT(14H     XWB       F10.2)
  560 FORMAT(14H     XDLI      F10.2)
  590 FORMAT(14H     XDLIENG   F10.2)
  600 FORMAT( 7H     XL  ,7X,F10.2,71H   XL CAN BE NEGATIVE.  IF SO, FUNCT
     13 WILL SET APPROPRIATE VALUES TO 0.)
  610 FORMAT(14H     XTHETST   F10.2)
  620 FORMAT(14H     XCAPENG   F10.2)
  670 FORMAT(14H     XATS1     F10.2)
  680 FORMAT(14H     XATS2     F10.2)
  690 FORMAT(14H     XMPPLAT   F10.2)
  700 FORMAT(14H     XATS3     F10.2)
  710 FORMAT(14H     XATS4     F10.2)
  720 FORMAT(14H     XPSA      F10.2)
  730 FORMAT(14H     XD        F10.2)
  740 FORMAT(14H     XPDASW    F10.2)
  750 FORMAT(14H     XSURS1    F10.2)
  755 FORMAT(14H     XSTG1     F10.2)
  760 FORMAT(14H     XSURS2    F10.2)
  770 FORMAT(14H     XTOTE     F10.2)
  780 FORMAT(14H     XSURS3    F10.2)
  790 FORMAT(14H     XSALVS    F10.2)
  800 FORMAT(14H     XPST      F10.2)
  840 FORMAT(14H     XEFFCM    F10.2)
C
C*    START ASW PORTION OF CTFMOD
```

```
C
      XD = (1./BAPL) * (BAREAL*UBASWL*XASWL + BAREA*UBASW*XASW*XEFFCM)
      PRINT 730, XD
C
      XPDASW = 1. - EXP(-ASWF*XD)
      PRINT 740, XPDASW
C
      XSURS1 = (1.-XPDASW*PKASW) * ST
      PRINT 750, XSURS1
C
      XSTG1=(1.-XPDASW*PKASW)*STG
      PRINT 755,XSTG1
C
      XSURS2 = XSURS1*(1.-PDIN*PKIN)
     X         + PKIN * FUNCT5(PDIN*XSURS1-ZLAMPF*XEASWA)
      PRINT 760, XSURS2
C
      XTOTE =(XEASWA + XEASWN)*(360./THSECT)
      PRINT 770, XTOTE
C
      XSURS3 = XSURS2*(1.-PKIIN * FUNCT6(XTOTE,ESLR,ESR,SUBSCR))
      PRINT 780,XSURS3
C*
      RS(1,L) = RS(1,L) - (ST-XSURS3)
      RS(2,L) = RS(2,L) - (STG-XSTG1)
      PRINT 810, RS(1,L),RS(2,L),L
  810 FORMAT(3X,9H RS(1,L)=,F14.4,9H RS(2,L)=,F14.4,8H FOR L =,I2)
C
      XSALVS=XSURS3*STSALV
      PRINT 790,XSALVS
C
      WNSHT=WFTAAW*XEAAW+WFTASW*(XEASWA+XEASWN)+WFTPLT*XPLAT+
     1       WFTURG*XURGS
      IF(WNSHT .LE. 0.) WNSHT=1
      IF(XPLAT.LE.0.) GO TO 950
      XSPPLA  = WFTPLT*XPLAT/WNSHT
      XSPPLA  =(XSALVS*XSPPLA)/XPLAT
C
      IXSP=XSPPLA
      FIXSP=IXSP
      FR=XSPPLA -FIXSP
      SMODP=XPLAT*FR
      XPST=(SMODP* FUNC12(TPS*(FIXSP+1.),TAB12)
     X     + (XPLAT-SMODP) *  FUNC12(FIXSP*TPS,TAB12))/XPLAT
      PRINT 800, XPST
C*
  950 CONTINUE
C*
      IF(ATT.GT.0. .OR. STG.GT.0.) GO TO 890
C*
      XEFFCM = XEFFCM*XPST
      PRINT 840, XEFFCM
C*
      PMAAW = 0.
      IF(XEAAW.GT.0.) GO TO 932
      PTAAW = 0.
      GO TO 934
```

```
  932 PTAAW = WFTAAW*XEAAW/WNSHT
      PTAAW = XSALVS*TPS*PTAAW/XEAAW
  934 XEASW = XEASWA + XEASWN
      PMASW = 0.
      IF(XEASW.GT.0.) GO TO 936
      PTASW = 0.
      GO TO 938
  936 PTASW = WFTASW*XEASW/WNSHT
      PTASW = XSALVS*TPS*PTASW/XEASW
  938 PMURG = 0.
      IF(XURGS.GT.0.) GO TO 940
      PTURG = 0.
      GO TO 942
  940 PTURG = WFTURG*XURGS/WNSHT
      PTURG = XSALVS*TPS*PTURG/XURGS
  942 CONTINUE
C*
      SURAAW = AMAX1(0.,(1.-PTAAW/HRTAAW))
      SURASW = AMAX1(0.,(1.-PTASW/HRTASW))
      SURURG = AMAX1(0.,(1.-PTURG/HRTURG))
C*
      XEAAW  = XEAAW  * SURAAW
      XEASWA = XEASWA * SURASW
      XEASWN = XEASWN * SURASW
      XURGS  = XURGS  * SURURG
C*
      PRINT 850, PMAAW,PMASW,PMURG,PTAAW,PTASW,PTURG
      PRINT 860, SURAAW,SURASW,SURURG
      PRINT 870, XEAAW,XEASWA,XEASWN,XURGS
C*
      GO TO 2000
C*
  890 CONTINUE
C*
C*    END ASW PORTION OF CTEMOD
C*
      XAEWST  = UBAEWL * XAEWL  + UBAEW * XAEW
      PRINT 510,  XAEWST
C
      XCAPST  =  CAPML * UBAEWL * XAEWL  +  CAPM * UBAEW*XAEW
      XFGTRA = AMIN1(XFGHTR,FGHTRI*XEFFCM)
      TEMP=XCAPST
      XCAPST=0.
      IF(BUCAP .GT. 0.) XCAPST=AMIN1(TEMP,XFGTRA/BUCAP)
      PRINT 520, XCAPST
      FUNCT = FUNCT2(XAEWST ,AEWD,STAR,THSFCA)
C
      XZ =  .5 * (  FUNCT + AEWD + STAR )
      PRINT 530, XZ
C*
      XZ = (1.-PRWLND)*XZ + PRWLND*AMAX1(XZ,WRRLND)
      PRINT 530, XZ
C*
      DO 1000 I=1,2
C*
      PRINT 621, I
  621 FORMAT(22H0START OF ITERATION I=,I2,36H THROUGH ATTRITION PORTION
```

```
          1OF CTFMOD)
C
          XCAPET = 0.
          XDLIET = 0.
          DO 900 K=1,NKRB
          AT(K) = BMR(1,K) + BMR(2,K)
          AT1(K) = 0.
      900 ATKT(K) = 0.
          AEST = AESC
          AESCKT = 0.
          ATKTT=0.
          FGHTRK = 0.
          FGHTRL = 0.
          AESCK  = 0.
          ATK    = 0.
C*
          DO 920 K=1,NKRB
C*
          ATT = 0.
          DO 902 KK=K,NKRB
      902 ATT = ATT + AT(KK)
          PRINT 32, ATT
          IF(ATT.LE.C. .AND. K.EQ.1) GO TO 645
          IF(ATT.LE.C) GO TO 922
C*
          XTB = XZ/VBT(K) - (D1T(I,K)/VBT(K)+(D2T(I,K)-SMALLR)/VI)
          PRINT 540, XTB
C
          XWB = FUNCT1(XTB,T1,T2,T3,T4)
          PRINT 550, XWB
C
          XDLI = XWB * WVSIZ
          XDLJ = AMAX1(0.0,XFGTRA-BUCAP*XCAPST-FTSORU)
          PRINT 560, XDLI
          PRINT 560, XDLJ
C
          XDLIEN = AMIN1(XDLJ*DLIA,XDLI)
          PRINT 590, XDLIEN
C
          XL = (VCAP/VBT(K))*XZ + CAPMR - VCAP*(TCAP+(D1T(I,K)/VBT(K)))
          PRINT 600,XL
C
          XTHETS = FUNCT3(XL,CAPST,D1T(I,K))
          PRINT 610,XTHETS
C
          TWOPI = 2. * 3.14159265
C
          XCAPEN  = XCAPST   *  (XTHETS  / TWOPI )*(360./THSECA)
          XCAPEN = AMIN1(XCAPEN,XCAPST)
          PRINT 62C, XCAPEN
C
          XCAPEN = AMAX1(0.,XCAPEN-XCAPET)
          XDLIEN = AMAX1(0.,XDLIEN-XDLIET)
          XCAPET = XCAPET + XCAPEN
          XDLIET = XDLIET + XDLIEN
          PRINT 59C, XDLIEN
          PRINT 62C, XCAPEN
```

SUBROUTINE CTEMCD

```
C*
      CALL ATRTIA(XDLIEN,XCAPEN,AEST,ATT,ZMPDLI,ZMPCAP,ZMPESC,PKAT1,PKDF
     11,FGHTRL,AESCK,ATK,IATRIA)
C*
      FGHTRK = FGHTRK + FGHTRL
      DO 904 KK=K,NKRB
      TATK = ATK*(AT(KK)/ATT)
      AT(KK) = AT(KK) - TATK
      ATKT(KK) = ATKT(KK) + TATK
      PRINT 623, KK,TATK
  623 FORMAT(8H FOR KK=,I2, 8H,   TATK=,F10.4)
  904 CONTINUE
      PRINT 624, (J,ATKT(J),J=1,NKRB)
  624 FORMAT(7(6H ATKT(,I1,2H)=,F10.4))
      PRINT 625, (J,AT(J),J=1,NKRB)
  625 FORMAT(7(6H    AT(,I1,2H)=,F10.4))
      AT1(K) = AT(K)
      ATKTT = ATKTT + ATKT(K)
      PRINT 630, K,AT1(K),K,ATKTT
  630 FORMAT(8H  FOR K=,I2,10H,   AT1(K)=,F10.4,23H -- SO FAR (THROUGH K
     1=,I2,8H) ATKTT=,F10.4)
      AESCKT = AESCKT + AESCK
      AEST=AMAX1(0.,AEST-AESCK-XDLIEN-XCAPEN)
      PRINT 631, AEST,AESCKT
  631 FORMAT(10X,10H      AEST=,F10.4,25X,8H AESCKT=,F10.4)
      PRINT 632, FGHTRL,FGHTRK
  632 FORMAT(10X,10H    FGHTRL=,F10.4,25X,8H FGHTRK=,F10.4)
C*
  920 CONTINUE
  922 CONTINUE
C*
      AESCK = AESCKT
      PRINT 640, AESCK
  640 FORMAT(8H  AESCK=,F10.4)
  645 CONTINUE
C*
      IF(I.EQ.2) GO TO 1000
C*
      PRINT 650
  650 FORMAT(38H0DISPLAY RESULTS OF ASM-VS-SHIP BATTLE)
C*
      XATST = 0.
      DO 924 K=1,NKRB
  924 XATST = XATST + AT1(K)*ZMPATT(K)
      XATS1 = XATST + ZMPSTG*XSTG1
      PRINT 670, XATS1
C*
      IF(XATS1.LE.0.) XATS1 = 1.
      PKPLD = 0.
      PKSS = 0.
      ENACD = 0.
      DO 926 ITAB=1,20
      TAB10(ITAB) = 0.
      TAB13(ITAB) = 0.
  926 CONTINUE
      DO 927 K=1,NKRB
      FMPBT(K) = (AT1(K)*ZMPATT(K))/XATS1
```

```
  927 CONTINUE
      NKRBP1 = NKRB+1
      FMRBT(NKRBP1) = ZMPSTG*XSTG1/XATS1
      DO 930 K=1,NKRBP1
      PKPLD = PKPLD + PKPLDT(K)*FMRBT(K)
      PKSS = PKSS + PKSST(K)*FMRBT(K)
      ENACD=ENACD+ ENACDT(K)*FMRBT(K)
      DO 928 ITAB=1,20
      TAB10(ITAB) = TAB10(ITAB) + TAB10T(ITAB,K)*FMRBT(K)
      TAB13(ITAB) = TAB13(ITAB) + TAB13T(ITAB,K)*FMRBT(K)
  928 CONTINUE
  930 CONTINUE
C*
      PRINT 675, (K,FMRBT(K),K=1,NKRB)
  675 FORMAT(6(7H FMRBT(,I1,2H)=,F10.4))
      PRINT 676, FMRBT(NKRBP1)
  676 FORMAT(15H FMRBT(NKRB+1)=,F10.4)
      PRINT 250
      PRINT 10, PKSS,FPPL1,PKPL1,PKPL2,PKPLD,FPPL2
      PRINT 310
      PRINT 12,(TAB10(ITAB),ITAB=1,20)
      PRINT 330
      PRINT 12,(TAB13(ITAB),ITAB=1,20)
C
      XATS2= (1.-PKSS)*XATS1 + PKSS*FUNCT5(XATS1-FUNCT9(XEAAW,TAB10))
      PRINT 680, XATS2
C
      WNSHM = WFMAAW*XEAAW+WFMASW*(XFASWA+XEASWN)+WFMPLT*XPLAT+
     1        WFMURG*XURGS
      IF(WNSHM .LE. 0.) WNSHM=1
      IF(XPLAT.LE.0.) GO TO 960
      XMPPLA  = WFMPLT*XPLAT/WNSHM
      XMPPLA  =(XATS2*XMPPLA)/XPLAT
      PRINT 690, XMPPLA
C
      XATS3 = (1.-PKPL1)*XMPPLA  + PKPL1*FUNCT5(XMPPLA -FPPL1)
      XATS3 = XATS3*XEFFCM + XMPPLA*(1.-XEFFCM)
      PRINT 700, XATS3
C
      XATS4 = XATS3 *(1.-PKPLD)*(1.-PKPL2)
     X        +(1.-PKPLD)*PKPL2* FUNC11(XATS3,FPPL2)
      XATS4 = XATS4*XEFFCM + XATS3*(1.-XEFFCM)
      PRINT 710,XATS4
C*
      TIACFT = ATTCKI + FGHTRI
      IF(TIACFT.LE.0.) GO TO 958
      PIACD = ENACD/TIACFT
      FACD = 1.-(1.-PIACD)**XATS4
      FDMCV = (XFGHTR-XCAPST-XDLIET)*FACD
      ADMCV = XATTCK*FACD
      XFGHTR = XFGHTR - FDMCV
      XATTCK = XATTCK - ADMCV
C*
      PRINT 844, ENACD,PIACD,FACD
      PRINT 845, FDMCV,XFGHTR
      PRINT 846, ADMCV,XATTCK
  844 FORMAT (8H  ENACD=,F10.4,8H  PIACD=,F10.4,8H   FACD=,F10.4)
```

```
  845 FORMAT (8H   FDMCV=,F10.4,8H XFGHTR=,F10.4)
  846 FORMAT (8H   ADMCV=,F10.4,8H XATTCK=,F10.4)
  958 CONTINUE
C
      XPSA =  FUNC12(XATS4,TAB13)
      PRINT 720,XPSA
C
      XEFFCM = XEFFCM*XPSA*XPST
      PRINT 840,XEFFCM
C
  960 CONTINUE
C*
      IF(XEAAW.GT.0.) GO TO 962
      PMAAW = 0.
      PTAAW = 0.
      GO TO 964
  962 PMAAW = WFMAAW*XEAAW/WNSHM
      PTAAW = WFTAAW*XEAAW/WNSHT
      PMAAW = (XATS2*PMAAW/XEAAW)*(1.-SSDAAW)
      PTAAW = XSALVS*TPS*PTAAW/XEAAW
  964 XEASW = XEASWA + XEASWN
      IF(XEASW.GT.0.) GO TO 966
      PMASW = 0.
      PTASW = 0.
      GO TO 968
  966 PMASW = WFMASW*XEASW/WNSHM
      PTASW = WFTASW*XEASW/WNSHT
      PMASW = (XATS2*PMASW/XEASW)*(1.-SSDASW)
      PTASW = XSALVS*TPS*PTASW/XEASW
  968 IF(XURGS.GT.0.) GO TO 970
      PMURG = 0.
      PTURG = 0.
      GO TO 972
  970 PMURG = WFMURG*XURGS/WNSHM
      PTURG = WFTURG*XURGS/WNSHT
      PMURG = (XATS2*PMURG/XURGS)*(1.-SSDURG)
      PTURG = XSALVS*TPS*PTURG/XURGS
  972 CONTINUE
C*
      SURAAW=AMAX1(0.,(1.-PMAAW/HRMAAW))*AMAX1(0.,(1.-PTAAW/HRTAAW))
      SURASW=AMAX1(0.,(1.-PMASW/HRMASW))*AMAX1(0.,(1.-PTASW/HRTASW))
      SURURG=AMAX1(0.,(1.-PMURG/HRMURG))*AMAX1(0.,(1.-PTURG/HRTURG))
C*
      XEAAW = XEAAW * SURAAW
      XEASWA= XEASWA* SURASW
      XEASWN= XEASWN* SURASW
      XURGS = XURGS * SURURG
C*
      PRINT 850,PMAAW,PMASW,PMURG,PTAAW,PTASW,PTURG
  850 FORMAT(8H    PMAAW,F10.4,8H    PMASW,F10.4,8H    PMURG,F10.4,8H    PTA
     1AW,F10.4,8H    PTASW,F10.4,8H    PTURG,F10.4)
      PRINT 860,SURAAW,SURASW,SURURG
  860 FORMAT(8H   SURAAW,F10.4,8H   SURASW,F10.4,8H   SURURG,F10.4)
      PRINT 870, XEAAW,XEASWA,XEASWN,XURGS
  870 FORMAT(8H    XEAAW,F10.4,8H   XEASWA,F10.4,8H   XEASWN,F10.4,8H    XUR
     1GS,F10.4)
C*
```

```
 1000 CONTINUE
C*
      PRINT 1001
 1001 FORMAT(38HODISPLAY RESULTS OF AIR-TO-AIR  BATTLE)
      AESC = AESC - AESCK
      XFGHTR = XFGHTR - FGHTRK
C*
      PRINT 1002, AESC,XFGHTR
 1002 FORMAT(8H   AESC=,F10.4,8H XFGHTR=,F10.4)
      PRINT 1003, (K,AT(K),K=1,NKRB)
 1003 FORMAT(5H  AT(,I1,2H)=,F10.4)
C*
      TOTSOR = XDLIET
      IF(TOTSOR.LE.0.) GO TO 1004
      TMPSOR = XDLIET - (FGHTRK*(XDLIET/TOTSOR))
      FTSORU = FTSORU + TMPSOR*PFFCNF
 1004 CONTINUE
      PRINT 1005, FTSORU,ATSORU
 1005 FORMAT(9H  FTSORU=,F10.4,9H  ATSORU=,F10.4)
C*
      DO 1006 KRB=1,NKRB
      TOT = BMR(1,KRB) + BMR(2,KRB)
      IF(TOT.LE.0.) TOT=1.
      DO 1006 I=1,2
      BMK = ATKT(KRB)*BMR(I,KRB)/TOT
      ATABT(I,KRB) = ATABT(I,KRB) + BMR(I,KRB) - BMK
 1006 CONTINUE
      TOT = ESC(1) + ESC(2)
      IF(TOT.LE.C.) TOT=1.
      DO 1010 I=1,2
      ESK = AESCK*ESC(I)/TOT
      AESCAB(I) = AESCAB(I) + ESC(I)-ESK
 1010 CONTINUE
C*
      DO 1012 KRB=1,NKRB
      PRINT 1011,ATABT(1,KRB),ATABT(2,KRB),KRB
 1011 FORMAT(14H ATABT(1,KRB)=,F10.4,14H ATABT(2,KRB)=,F10.4,10H FOR KRB
     1 =,I2)
 1012 CONTINUE
      PRINT 1013,AESCAB(1),AESCAB(2)
 1013 FORMAT(14H    AESCAB(1)=,F10.4,14H    AESCAB(2)=,F10.4)
C*
 2000 CONTINUE
C
      WRITE(6,5990) XEFFCM
 5990 FORMAT(44H RELATIVE CARRIER CAPABILITY (XEFFCM) IS NOW,F7.4,1H.)
      WRITE(6,5999)
      WRITE(6,2)
 5999 FORMAT(25H END OF SUBROUTINE CTFMOD)
    2 FORMAT(51H --------------------------------------------------)
C
 3000 CONTINUE
C
      RETURN
      END
```

FUNCTION FUNCT1

```
C* DECK FUNCT1
      FUNCTION FUNCT1 (X,T1,T2,T3,T4)
      TERM1= (X-T1-T2-T3)/T4
      IF (TERM1-0.) 10, 20, 20
   10 FUNCT1=0.
      RETURN
   20 ITERM1=TERM1
      TERM2= ITERM1
      FUNCT1 = TERM2+1.
      RETURN
      END
```

FUNCTION FUNCT2

```
C* DECK FUNCT2
      FUNCTION FUNCT2(X,AEWD,STAR,THSECA)
      X = X*(360./THSECA)
      IF(X .LE. 0.) GO TO 90
      DIF=AEWD-STAR
      IF(DIF .GE. 0.) GO TO 5
      IF(X-2)90,90,7
    5 IF(X-2) 20,20,8
    7 THETA=3.14159265/X
      DIST=STAR*SIN(THETA)
      DIFF=AEWD-DIST
      IF(DIFF)90,90,10
    8 THETA=3.14159265/X
      DIST=STAR*SIN(THETA)
      DIFF=AEWD-DIST
   10 PROD=DIFF*(AEWD+DIST)
      TERM=(STAR-DIST)*(STAR+DIST)
      FUNCT2=SQRT(TERM)+SQRT(PROD)
      GO TO 100
   20 IF(X .LE. 1.) GO TO 80
      PROD=DIF*(AEWD+STAR)
      FUNCT2=(X-1.)*SQRT(PROD)+(2.-X)*DIF
      GO TO 100
   80 FUNCT2=X*DIF
      GO TO 100
   90 FUNCT2 = 0.
  100 CONTINUE
      RETURN
      END
```

FUNCTION FUNCT3

```
C* DECK FUNCT3
      FUNCTION FUNCT3(X,CAPSTR,D1)
      TERM1 = ABS(CAPSTR -D1)
      TERM2 = SQRT( CAPSTR **2 + D1**2)
      TERM3 = CAPSTR  + D1
      IF(X-TERM1) 10,3,3
    3 IF(X-TERM2) 20,20,4
    4 IF(X-TERM3)30,30,40
C
   10 FUNCT3=0.
      RETURN
C
   20 TERM4 = (CAPSTR **2 + D1**2 - X**2) / (2.*D1)
      TERM5 = (1./ CAPSTR ) * SQRT ( CAPSTR **2 - TERM4**2)
      TERM6 = ASIN(TERM5)
      FUNCT3 = 2. * TERM6
      RETURN
C
   30 TERM4 = (CAPSTR **2 + D1**2 - X**2) / (2.*D1)
      TERM5 = (1./CAPSTR ) * SQRT (CAPSTR **2 - TERM4**2)
      TERM6=ASIN(TERM5)
      FUNCT3 = (2.*3.14159265) - (2.* TERM6)
      RETURN
C
   40 FUNCT3 = 2.* 3.14159265
      RETURN
C
      END
```

FUNCTION FUNCT5

```
C* DECK FUNCT5
      FUNCTION FUNCT5(X)
      TERM=X
      IF(TERM-0.) 10,10,20
   10 FUNCT5=0.
      RETURN
   20 FUNCT5=TERM
      RETURN
      END
```

FUNCTION FUNCT6

```
C* DECK FUNCT6
      FUNCTION FUNCT6(X,ESLR,ESR,SUBSOR)
      TERM1=ESR+ESLR
      IF(ESR.GT.ESLR) GO TO 1
      TERM2=0.
      GO TO 2
    1 CONTINUE
      TERM2=SQRT(ESR**2-ESLR**2)
    2 CONTINUE
      IF(SUBSOR-TERM1)3,10,10
    3 IF(SUBSOR-TERM2)20,20,30
C
   10 FUNCT6=0.
      RETURN
C
   20 ANGLE1=ACOS(TERM2/ESR)
      TERM3=(X*ANGLE1)/3.14159265
      FUNCT6=AMIN1(1.,TERM3)
      RETURN
C
   30 TERM4=(ESR**2-ESLR**2+SUBSOR**2)/(2.*ESR*SUBSOR)
      ANGLE2=ACOS(TERM4)
      TERM5=(X*ANGLE2)/3.14159265
      FUNCT6=AMIN1(1.,TERM5)
      RETURN
C
      END
```

```
C* DECK FUNCT9
      FUNCTION FUNCT9(Y,TAB10)
      DIMENSION TAB10(20)
      ITERM1=Y
      TERM1=ITERM1
      TERM2=TERM1+1.
      FUNCT9=(Y-TERM1)* FUNC10(TERM2,TAB10)+(TERM2-Y) *FUNC10(TERM1,TAB1
     10)
      RETURN
      END
```

FUNCTION FUNC10

```
C* DECK FUNCT10
      FUNCTION FUNC10(X,TAB10)
      DIMENSION TAB10(20)
      IF(X .GE. 1.) GO TO 3
      FUNC10=0.
      RETURN
    3 CONTINUE
      IX1=X
      IX2=IX1+1
      X1=IX1
      FUNC10 =TAB10(IX1)+(TAB10(IX2)-TAB10(IX1))*(X-X1)
      RETURN
      END
```

FUNCTION FUNC11

```
      C* DECK FUNCT11
            FUNCTION FUNC11(X,FPPL2)
            TERM=X-FPPL2
            IF(TERM-0.)10,10,20
         10 FUNC11 =0.
            RETURN
         20 FUNC11 = TERM
            RETURN
            END
```

FUNCTION FUNC12

```
C* DECK FUNCT12
      FUNCTION FUNC12(X,TAB12)
      DIMENSION TAB12(20)
      IF(X-0.)5,5,8
    5 FUNC12 =1.
      RETURN
    8 IF(X-10.)10,10,20
   10 IX1=X
      IX2=IX1+1
      X1=IX1
      IF(X-1.)15,18,18
   15 FUNC12  = 1. -(1.-TAB12(IX2))*(X-X1)
      RETURN
   18 FUNC12 =TAB12(IX1) - (TAB12(IX1) - TAB12(IX2)) * (X-X1)
      RETURN
   20 FUNC12 =0.
      RETURN
      END
```

SUBROUTINE DDAY

```
C* DECK DDAY
      SUBROUTINE DDAY(L)
C*
C*    DDAY MODELS THE DDAY SHOOTOUT
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNOS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB1OT(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAFW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE DDAY

```
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
      DIMENSION RSKBA(10),RSAC(10),RSAS(10),RSKAA(10),RSNEW(10),BSK(5)
      SURV2(A,T,P)=T*(1.-((A/T)-AINT(A/T))*P)*((1.-P)**INT(A/T))
      WRITE(6,1)
      WRITE(6, 201)
    1 FORMAT(51H0--------------------------------------------------)
  201 FORMAT(24H START SUBROUTINE DDAY   )
      WRITE(6,202)  L
  202 FORMAT(54H RESULTS OF THE DDAY SHOOTOUT--TASK FORCE IS IN REGION ,
     1I3)
      BSK(1)=0.
C
C  RED ATTACKING SHIPS
C
      DO 5 KRS=1,NKRS
      RSKBA(KRS)=AMIN1(DDRKBA(KRS),RS(KRS,L))
      RSS=RS(KRS,L)-RSKBA(KRS)
      RSA=AMIN1(RSS,DDRSA(KRS))
      RSAC(KRS)=RSA*DDFAC(KRS)
      RSAS(KRS)=RSA-RSAC(KRS)
    5 CONTINUE
      TC=XPLAT
      TS=XEAAW+XEASWA+XEASWN+XURGS
C
C  COMPUTE DEGRADATION OF CARRIER CAPABILITY
C
      ATT=0.
      IF(TC .EQ. 0.) GO TO 20
      IF(IDDAC .EQ. 2) GO TO 15
      PROD=1.
      DO 11  KRS=1,NKRS
      SHOTS=RSAC(KRS)*DDSPA(KRS)
      ATT=ATT+SHOTS
      IF(SHOTS .EQ. 0.) GO TO 11
      TERM=DDPKC(KRS)/AMAX1(1.,TC)
      XIER=(1.-TERM)**SHOTS
      PROD=PROD*XIER
   11 CONTINUE
      XEFFCM=PROD
      GO TO 20
   15 ATT=0.
      SUM=0.
      DO 16 KRS=1,NKRS
      SHOTS=RSAC(KRS)*DDSPA(KRS)
      ATT=ATT+SHOTS
      SUM=SUM+SHOTS*DDPKC(KRS)
   16 CONTINUE
      AVPK=SUM/ATT
      IF(ATT .LE. TC) GO TO 17
      SURV=SURV2(ATT,TC,AVPK)
      GO TO 18
   17 SURV=TC-ATT*AVPK
   18 XEFFCM=SURV/TC
   20 CONTINUE
      CTT = ATT
```

```
C
C   COMPUTE ATTRITION TO OTHER BLUE SHIPS
C
      IF(TS .LE. .0001) GO TO 31
      IF(IDDAS .EQ. 2) GO TO 25
      PROD=1.
      DO 21 KRS=1,NKRS
      SHOTS=RSAS(KRS)*DDSPA(KRS)
      IF(SHOTS .EQ. 0.) GO TO 21
      TERM=DDPKS(KRS)/AMAX1(1.,TS)
      XIER=(1.-TERM)**SHOTS
      PROD=PROD*XIER
   21 CONTINUE
      TSK=TS*(1.-PROD)
      GO TO 30
   25 ATT=0.
      SUM=0.
      DO 26 KRS=1,NKRS
      SHOTS=RSAS(KRS)*DDSPA(KRS)
      ATT=ATT+SHOTS
      SUM=SUM+SHOTS*DDPKS(KRS)
   26 CONTINUE
      AVPK=SUM/ATT
      IF(ATT .LE. TS) GO TO 28
      TSK=TS-SURV2(ATT,TS,AVPK)
      GO TO 30
   28 TSK=AVPK*ATT
   30 CONTINUE
      FBSK=TSK/TS
      GO TO 32
   31 FBSK=0.
   32 CONTINUE
      BSK(2)= XEAAW*FBSK
      BSK(3)=XEASWA*FBSK
      BSK(4)=XEASWN*FBSK
      BSK(5)= XURGS*FBSK
C
C   RED SHIPS KILLED AFTER ATTACK ON TASK FORCE
C
      DO 35 KRS=1,NKRS
      RSS=RS(KRS,L)-RSKBA(KRS)
      RSKAA(KRS)=AMIN1(DDRKAA(KRS),RSS)
      RSNEW(KRS)=RSS-RSKAA(KRS)
   35 CONTINUE
C
C   COMPUTE ATTRITION TO AIRCRAFT ON CARRIERS
C
      IF(XPLAT .EQ. 0.) GO TO 46
      ENACD = 0.
      PIACD = 0.
      FACD  = 0.
      FDMCV = 0.
      ADMCV = 0.
      TIACFT = ATTCKI + FGHTRI
      IF(TIACFT.LE.0.) GO TO 46
      ATT = CTT - RSAC(1)*DDSPA(1)
      IF(ATT.LE.0.) GO TO 46
```

```
      FRAC    = RSAC(2)*DDSPA(2)/ATT
      NKRBP1 = NKRB+1
      ENACD = ENACDT(NKRBP1)*FRAC
      DO 42 KRS=3,NKRS
      FRAC       = RSAC(KRS)*DDSPA(KRS)/ATT
      ENACD = ENACD + ENACDS(KRS)*FRAC
   42 CONTINUE
   44 CONTINUE
      PIACD = ENACD/TIACFT
      PIACD = AMIN1(0.9999,PIACD)
      FACD = 1.-(1.-PIACD)**ATT
      FDMCV = XFGHTR*FACD
      ADMCV = XATTCK*FACD
      XFGHTR = XFGHTR - FDMCV
      XATTCK = XATTCK - ADMCV
   46 CONTINUE
C
C  OUTPUT RESULTS, UPDATING QUANTITIES AS NECESSARY
C
      WRITE(6,240) (KRS,KRS=1,NKRS)
      WRITE(6,241) (RS(KRS,L),KRS=1,NKRS)
      WRITE(6,242) (RSKBA(KRS),KRS=1,NKRS)
      WRITE(6,243) ( RSAC(KRS),KRS=1,NKRS)
      WRITE(6,244) ( RSAS(KRS),KRS=1,NKRS)
      WRITE(6,245) (RSKAA(KRS),KRS=1,NKRS)
      WRITE(6,246) (RSNEW(KRS),KRS=1,NKRS)
  240 FORMAT(17HOKIND OF RED SHIP,24X,10I8)
  241 FORMAT(28H INITIAL RED SHIPS IN REGION,13X,10F8.3)
  242 FORMAT(41H SHIPS KILLED BEFORE ATTACK ON TASK FORCE ,10F8.3)
  243 FORMAT(25H SHIPS ATTACKING CARRIERS,16X,10F8.3)
  244 FORMAT(33H SHIPS ATTACKING OTHER BLUE SHIPS, 8X,10F8.3)
  245 FORMAT(41H SHIPS KILLED AFTER ATTACK ON TASK FORCE ,10F8.3)
  246 FORMAT(30H RESULTANT RED SHIPS IN REGION,11X,10F8.3)
      DO 48 KRS=1,NKRS
      RS(KRS,L)=RSNEW(KRS)
   48 CONTINUE
      WRITE(6,250)
      WRITE(6,249)
      WRITE(6,251) XPLAT,XEAAW,XEASWA,XEASWN,XURGS
      WRITE(6,252) (BSK(I),I=1,5)
      XEAAW =XEAAW -BSK(2)
      XEASWA=XEASWA-BSK(3)
      XEASWN=XEASWN-BSK(4)
      XURGS =XURGS -BSK(5)
      WRITE(6,253) XPLAT,XEAAW,XEASWA,XEASWN,XURGS
  249 FORMAT(1H+,17HKIND OF BLUE SHIP )
      IF(XPLAT.GT.0.) WRITE(6,254) XEFFCM
  250 FORMAT(1H0,43X,37HXPLAT   XEAAW  XEASWA  XEASWN   XURGS )
  251 FORMAT(29H INITIAL BLUE SHIPS IN REGION,12X,5F8.3)
  252 FORMAT(18H BLUE SHIPS KILLED,23X,5F8.3)
  253 FORMAT(31H RESULTANT BLUE SHIPS IN REGION,10X,5F8.3)
  254 FORMAT(54H RESULTANT RELATIVE CARRIER CAPABILITY (XEFFCM) EQUALS,
     1F8.4,1H.)
      IF(XPLAT .EQ. 0.) GO TO 99
      PRINT 284, ENACD,PIACD,FACD
      PRINT 285, FDMCV,XFGHTR
      PRINT 286, ADMCV,XATTCK
```

```
  284 FORMAT (8H  ENACD=,F10.4,8H  PIACD=,F10.4,8H   FACD=,F10.4)
  285 FORMAT (8H  FDMCV=,F10.4,8H XFGHTR=,F10.4)
  286 FORMAT (8H  ADMCV=,F10.4,8H XATTCK=,F10.4)
   99 WRITE(6,299)
      WRITE(6,2)
  299 FORMAT(25H END OF SUBROUTINE DDAY  )
    2 FORMAT(51H --------------------------------------------------)
C
      RETURN
      END
```

SUBROUTINE GNAATK

```
C* DECK GNAATK
      SUBROUTINE GNAATK(L,ITP)
C*
C*    GNAATK GENERATES AIR ATTACKS ON THE TASK FORCE
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJD(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSDRR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAFGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE GNAATK

```
C*
C* COMDECK COMGA
      COMMON/COMGA/ NTPSLA,BMR(2,3),ESC(2)
C*
C*
      WRITE(6,1)
      WRITE(6, 201)
    1 FORMAT(51H0--------------------------------------------------)
  201 FORMAT(24H START SUBROUTINE GNAATK)
C*
      DO 10 IAB=1,2
      DO  8 KRB=1,NKRB
    8 BMR(IAB,KRB) = 0.
      ESC(IAB) = 0.
   10 CONTINUE
      BMT = 0.
      EST = 0.
      IF(NTPSLA.GT.0.AND.NTPSLA.LT.IATKRT(L)) GO TO 40
      DO 20 IAB=1,2
      DO 18 KRB=1,NKRB
   18 BMR(IAB,KRB)=AVAILT(L,IAB,KRB)*ATABT(IAB,KRB)
      ESC(IAB)=AVAILE(L,IAB)*AESCAB(IAB)
   20 CONTINUE
      DO 22 KRB=1,NKRB
   22 BMT = BMT + BMR(1,KRB) + BMR(2,KRB)
      EST = ESC(1)+ESC(2)
      IF(BMT.GE.BMTMIN(L)) GO TO 30
      WRITE(6,281)
  281 FORMAT(55H INSUFFICIENT RED BOMBERS. NO AIR ATTACK ON TASK FORCE.)
      DO 26 IAB=1,2
      DO 24 KRB=1,NKRB
   24 BMR(IAB,KRB) = 0.
      ESC(IAB) = 0.
   26 CONTINUE
      BMT = 0.
      EST = 0.
      NTPSLA=NTPSLA+1
      GO TO 100
   30 NTPSLA = 1
      GO TO 100
   40 NTPSLA = NTPSLA+1
      WRITE(6,283)
  283 FORMAT(55H NO RED AIR ATTACK ON TASK FORCE SCHEDULED THIS PERIOD.)
  100 CONTINUE
      PRINT 210,BMT,EST,NTPSLA,ITP
  210 FORMAT(8H     BMT=,F10.4,8H     EST=,F10.4,8H NTPSLA=,I5,5H ITP=,I5)
C*
      DO 219 I=1,2
      PRINT 212, I
  212 FORMAT(33H0THE FOLLOWING VALUES ARE FOR I =,I2)
      PRINT 213, ESC(I)
  213 FORMAT(10H    ESC(I)=,F10.4)
      PRINT 214, (K,BMR(I,K),K=1,NKRB)
  214 FORMAT(7H BMR(I,,I1,2H)=,F10.4)
      DO 215 KRB=1,NKRB
  215 ATABT(I,KRB) = ATABT(I,KRB) - BMR(I,KRB)
      AESCAB(I) = AESCAB(I) - ESC(I)
```

SUBROUTINE GNAATK

```
      PRINT 216, I
  216 FORMAT(34H AIRCRAFT ON GROUND ON AIRBASE I =,I2,3H --)
      PRINT 217, AESCAB(I)
  217 FORMAT(12H  AESCAB(I)=,F10.4)
      PRINT 218, (K,ATABT(I,K),K=1,NKRB)
  218 FORMAT(9H ATABT(I,,I1,2H)=,F10.4)
  219 CONTINUE
C*
      WRITE(6, 299)
      WRITE(6,2)
  299 FORMAT(25H END OF SUBROUTINE GNAATK)
    2 FORMAT(51H --------------------------------------------------)
C
      RETURN
      END
```

```
      SUBROUTINE MOVRS(LOCTF,ITP)
C
C  SUBROUTINE MOVRS MOVES RED SHIPS(SURFACE SHIPS AND SUBMARINES)
C  FROM REGION TO (ADJACENT) REGION AND ASSESSES BARRIER KILLS AND
C  COUNTERKILLS AS APPROPRIATE
C
C  MATRIX RS(KRS,LOC) CONTAINS THE NUMBER OF RED SHIPS OF KIND KRS IN
C  LOCATION LOC
C  LOCTF IS NEW LOCATION OF TASK FORCE, AFTER MOVTF HAS BEEN EXECUTED
C
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJD(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJD(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
```

SUBROUTINE MOVRS

```
      COMMON   WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON   WRLNDO(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON   XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON   XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON   ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C
C
      DIMENSION PEN(10),PENK(10),RSMO(10,5),RSMI(10,5),ZERO(10)
      DIMENSION RSNEW(10)
      DATA ZERO /10*0./
      WRITE(6,1)
      WRITE(6, 501)
    1 FORMAT(51H0--------------------------------------------------)
  501 FORMAT(24H START SUBROUTINE MOVRS )
      IHEAD=0
      DO 11 LOC=1,NLOC
      DO 3 KRS=1,NKRS
      RSMO(KRS,LOC)=0.
      RSMI(KRS,LOC)=0.
    3 CONTINUE
   11 CONTINUE
      DO 10 LOC=1,NLOC
      IF(LOC .EQ. LOCTF) GO TO 10
      NEWLOC= LOC+ (LOCTF-LOC)/IABS(LOCTF-LOC)
      IF(NEWLOC .EQ. 0 ) GO TO 10
      LOCTF1=LOCTF+1
      DO 4 KRS=1,NKRS
      RSMO(KRS,LOC)=RS(KRS,LOC)*PRSM(KRS,LOC,LOCTF1)
    4 CONTINUE
      IBAR=MAX0(LOC,NEWLOC)
      ICTLB=ICTL(IBAR)+1
      GO TO (6,5,6,5),ICTLB
    5 CONTINUE
      IF(IHEAD .EQ. 1) GO TO 14
      WRITE(6,510) ITP
  510 FORMAT(73H ATTRITION TO RED SHIPS TRANSITING BLUE-CONTROLLED BARRI
     1ERS DURING PERIOD ,I5,21H, BY KIND OF RED SHIP )
      IHEAD=1
   14 CONTINUE
      SIB=BSIBAR(IBAR)
      NKRP=NKRS
      DO 15 KRP=1,NKRP
      PEN(KRP)=RSMO(KRP,LOC)
   15 CONTINUE
      SP=0.
      DO 16 KRP=1,NKRP
      SP=SP+RACCDW(KRP)*RACPCK(KRP)
   16 CONTINUE
      IBCP=2
      IF(SP .EQ. 0.) IBCP=1
      IF(IBCP .EQ. 2) GO TO 20
      CALL BARKCK(BSIBAR(IBAR),BARLTH(IBAR),NKRP,PEN,BEDW,CPRPK,
     1RECDW,CPBSCK,AWRCBB,PENK,SIBCK)
      GO TO 30
   20 CONTINUE
      CALL BARKCK(BSIBAR(IBAR),BARLTH(IBAR),NKRP,PEN,ZERO,ZERO,
     1RACCDW,RACPCK,.5,PENK,SIBCK1 )
```

```
      SIB=BSIBAR(IBAR)-SIBCK1
      CALL BARKCK(SIB,BARLTH(IBAR),NKRP,PEN,BEDW,CPRPK,
     1RECDW,CPBSCK,AWRCBB,PENK,SIBCK2 )
      SIBCK=SIBCK1+SIBCK2
   30 CONTINUE
C
C     PRINT OUT KILLS
C
      IBARM1=IBAR-1
      WRITE(6,521) IBARM1,IBAR
      WRITE(6,522) (PEN(KRS),KRS=1,NKRS)
      WRITE(6,523) (PENK(KRS),KRS=1,NKRS)
      SUBNEW=BSIBAR(IBAR)-SIBCK
      WRITE(6,524) BSIBAR(IBAR), SIBCK, SUBNEW
  521 FORMAT(1H ,4X,23HBARRIER BETWEEN REGIONS ,I3,4H AND,I3)
  522 FORMAT(1H ,7X,28HRED SHIPS ATTEMPTING TRANSIT, 5X,10F7.2)
  523 FORMAT(1H ,7X,16HRED SHIPS KILLED,17X,10F7.2)
  524 FORMAT(1H ,7X,25HBLUE BARRIER SUBMARINES--,F7.2,15H INITIALLY LESS
     1,F7.2,21H COUNTERKILLED YIELDS,F7.2,11H SURVIVING. )
      GO TO 8
C
C     IF THAT BARRIER IS NOT PLAYED OR IS RED-CONTROLLED,NO ATTRITION
C
    6 SIBCK=0.
      DO 7 KRS=1,NKRS
      PENK(KRS)=0.
    7 CONTINUE
    8 BSIBAR(IBAR)=BSIBAR(IBAR)-SIBCK
      DO 9 KRS=1,NKRS
      RSMI(KRS,NEWLOC)=RSMI(KRS,NEWLOC)+RSMO(KRS,LOC)-PENK(KRS)
    9 CONTINUE
   10 CONTINUE
C
C     UPDATE NUMBER OF RED SHIPS, BY KIND AND LOCATION
C     OUTPUT RESULTS
C
      IF(IHEAD .EQ. 0) WRITE(6,540)
  540 FORMAT(83H NO BARRIERS ARE BLUE-CONTROLLED, HENCE THERE IS NO BARR
     1ER ATTRITION TO RED SHIPS.    )
      WRITE(6,550) ITP
  550 FORMAT(32H FLOW OF RED SHIPS DURING PERIOD ,I5,21H, BY KIND OF RED
     1 SHIP   )
      DO 12 LOC=1,NLOC
      DO 13 KRS=1,NKRS
      RSNEW(KRS)=RS(KRS,LOC)+RSMI(KRS,LOC)-RSMO(KRS,LOC)
   13 CONTINUE
      WRITE(6,551) LOC
      WRITE(6,552) (  RS(KRS,LOC),KRS=1,NKRS)
      WRITE(6,553) (RSMI(KRS,LOC),KRS=1,NKRS)
      WRITE(6,554) (RSMO(KRS,LOC),KRS=1,NKRS)
      WRITE(6,555) (   RSNEW(KRS),KRS=1,NKRS)
  551 FORMAT(1H 4X,6HREGION,I2)
  552 FORMAT(1H 7X,27HINITIAL RED SHIPS IN REGION,6X,10F7.2)
  553 FORMAT(1H 7X,25HRED SHIPS ENTERING REGION, 8X,10F7.2)
  554 FORMAT(1H 7X,24HRED SHIPS LEAVING REGION, 9X,10F7.2)
  555 FORMAT(1H 7X,33HRESULTANT RED SHIPS IN REGION    ,10F7.2)
      DO 17 KRS=1,NKRS
```

```
      RS(KRS,LOC)=RSNEW(KRS)
   17 CONTINUE
   12 CONTINUE
C
      WRITE(6, 599)
      WRITE(6,2)
  599 FORMAT(25H END OF SUBROUTINE MOVRS )
    2 FORMAT(51H --------------------------------------------------)
C
      RETURN
      END
```

SUBROUTINE MOVTF

```
C* DECK MOVTF
      SUBROUTINE MOVTF(LOCTF,ITP)
C
C  SUBROUTINE MOVTF MOVES THE BLUE TASK FORCE AND ASSESSES BARRIER
C  ATTRITION AS APPROPRIATE
C
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
```

SUBROUTINE MOVTF

```
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
      COMMON /BARSCK/ SCK31,SCK32
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C
      INTEGER DESTF
      DIMENSION PEN1(10),PEN2(10),PENK(10),PENKA(10),ZERO(10),BSITF(10)
      DIMENSION DGF1(2)
      DATA ZERO /10*0./
      WRITE(6,1)
      WRITE(6, 501)
    1 FORMAT(51H0--------------------------------------------------)
  501 FORMAT(24H START SUBROUTINE MOVTF )
  541 FORMAT(67H ATTRITION TO (BLUE) TASK FORCE IN CROSSING BARRIER BETW
     1EEN REGIONS,I3,4H AND,I3)
  542 FORMAT(1H ,4X,17HKIND OF BLUE SHIP,21X,40HXPLAT  XEAAW XEASWA XEAS
     1WN  XURGS BSSNDS)
  543 FORMAT(1H ,4X,32HINITIAL BLUE SHIPS IN TASK FORCE,4X,6F7.2)
  544 FORMAT(1H ,4X,28HBLUE SHIPS KILLED BY BARRIER,8X,6F7.2)
  545 FORMAT(1H ,4X,36HRESULTANT BLUE SHIPS IN TASK FORCE  ,6F7.2)
  546 FORMAT(1H ,4X,30HCARRIER CAPABILITY DEGRADED BY,F7.4,49H, NEW RELA
     1TIVE CARRIER CAPABILITY (XEFFCM) EQUALS,F7.4,1H.)
  547 FORMAT(25H RED BARRIER SUBMARINES--,F7.2,15H INITIALLY LESS,F7.2,
     121H COUNTERKILLED YIELDS,F7.2,11H SURVIVING.)
  599 FORMAT(25H END OF SUBROUTINE MOVTF )
    2 FORMAT(51H --------------------------------------------------)
C
  570 FORMAT(39H TASK FORCE DOES NOT MOVE DURING PERIOD,I4,23H.  IT REMA
     1INS IN REGION,I3,1H.)
  571 FORMAT(14H DURING PERIOD,I4,29H TASK FORCE MOVES FROM REGION,I3,
     110H TO REGION,I3,1H.)
  576 FORMAT(24H BARRIER BETWEEN REGIONS,I3,4H AND,I3,   70H IS CONTROLL
     1ED BY BLUE, HENCE THERE IS NO ATTRITION TO THE TASK FORCE.)
  577 FORMAT(36H THERE IS NO BARRIER BETWEEN REGIONS,I3,4H AND,I3,48H, H
     1ENCE THERE IS NO ATTRITION TO THE TASK FORCE.)
      NKBS=6
C
C  FILL VECTOR BSITF WITH VALUES FROM BLANK COMMON
C
      BSITF(1)=XPLAT
      BSITF(2)=XEAAW
      BSITF(3)=XEASWA
      BSITF(4)=XEASWN
      BSITF(5)=XURGS
      BSITF(6)=BSSNDS
C
C  FIND NEW TASK FORCE LOCATION AND ASSESS NECESSITY FOR ATTRITION
C  COMPUTATIONS
C
      ITPP1=ITP+1
      DESTF=LOCTFF(ITPP1,LGTHMP,LTFMP,MIMP)
      IF(DESTF .EQ. LOCTF) GO TO 3
      IF(IABS(DESTF-LOCTF) .GE. 2) GO TO 97
      IF(DESTF .GT. NLOC) GO TO 98
      IBAR=MAX0(DESTF,LOCTF)
      GO TO 4
```

```
    3 CONTINUE
      WRITE(6,570) ITP,LOCTF
      WRITE(6,599)
      WRITE(6,2)
      RETURN
    4 CONTINUE
      WRITE(6,571) ITP,LOCTF,DESTF
      IF(ICTL(IBAR)-1) 50,51,5
   50 WRITE(6,577) LOCTF,DESTF
      GO TO 55
   51 WRITE(6,576) LOCTF,DESTF
   55 CONTINUE
      LOCTF=DESTF
      WRITE(6,599)
      WRITE(6,2)
      RETURN
    5 CONTINUE
      WRITE(6,541) LOCTF,DESTF
      IF(XPLAT .EQ. 0.) GO TO 6
      IBCP=3
      GO TO 8
    6 SP=0.
      DO 7 KBS=1,NKBS
      SP=SP+BACCDW(KBS)*BACPCK(KBS)
    7 CONTINUE
      IBCP=2
      IF(SP .EQ. 0.) IBCP=1
C
C     COMPUTE BARRIER ATTRITION
C
    8 IF(IBCP-2) 10,20,30
   10 CALL BARKCK(RSIBAR(IBAR),BARLTH(IBAR),NKBS,BSITF,REDW,CPBPK,
     1BECDW,CPRSCK,ATTWGT,PENK,SIBCK)
      GO TO 40
   20 CALL BARKCK(RSIBAR(IBAR),BARLTH(IBAR),NKBS,BSITF,ZERO,ZERO,
     1BACCDW,BACPCK,.5,PENK,SIBCK1)
      SIB=RSIBAR(IBAR)-SIBCK1
      CALL BARKCK(SIB,BARLTH(IBAR),NKBS,BSITF,REDW,CPBPK,
     1BECDW,CPRSCK,ATTWGT,PENK,SIBCK2)
      SIBCK=SIBCK1+SIBCK2
      GO TO 40
   30 WTFCB=AMIN1(WTFCBO,BARLTH(IBAR))
      SIB=RSIBAR(IBAR)*(WTFCB/BARLTH(IBAR))
      BACCDW(1)=CACDWO*XEFFCM
      CALL BARKCK(SIB,WTFCB,NKBS,BSITF,ZERO,ZERO,
     1 BACCDW,BACPCK,.5,PENK,SIBCK1)
      SIB=SIB-SIBCK1
      FM3(1) =0.
      DO 31 KBS=1,NKBS
      PEN1(KBS)=BSITF(KBS)*FM3(KBS)
      PEN2(KBS)=BSITF(KBS)-PEN1(KBS)
   31 CONTINUE
      CALL BARKCK(SIB,WTFCB,NKBS,PEN1,REDW,CPBPK,
     1BECDW,CPRSCK,ATTWGT,PENK ,SIBCK2)
      SIB=SIB-SIBCK2
      BECDW(1)=CSCDWO*XEFFCM
      CPBPK(1)=0.
```

```
      CALL BARKCK(SIB,WTFCB,NKBS,PEN2,REDW,CPBPK,
     1BECDW,CPRSCK,ATTWGT,PENKA,SIBCK3)
C  THIS COMPUTES COUNTERKILLS TO RED BARRIER SUBMARINES AND KILLS TO
C  ALL SHIPS EXCEPT CARRIERS
      DO 32 KBS=1,NKBS
      PENK(KBS)=PENK(KBS)+PENKA(KBS)
   32 CONTINUE
      SIBCK=SIBCK1+SIBCK2+SIBCK3
C
C  COMPUTE PROPORTION OF CARRIER CAPABILITY DESTROYED BY
C  RED BARRIER SUBMARINES
C
      ITER=1
   33 CONTINUE
      SCR=SIB*AMIN1(WTFCB,REDW(1))/WTFCB
      NSCR=SCR
      XSCR=NSCR
      FSCR=SCR-XSCR
      SUM=0.
      NSCRP1=NSCR+1
      DO 34 INDEX=1,NSCRP1
      NAS=INDEX-1
      AS=NAS
      TERM1=BINOM(NSCR,NAS,CPAGV)
      TERM2=TERM1*((XSCR+1.)/(XSCR-AS+1.))*(1.-CPAGV)
C   IN FACT, TERM2=BINOM(NSCR+1,NAS,CPAGV)
      TERM=TERM1*(1.-FSCR) + TERM2*FSCR
      SUM=SUM+FUNC12(AS*TPAS,TAB12)*TERM
   34 CONTINUE
   35 CONTINUE
      DGF1(ITER)=SUM+FSCR*(CPAGV**(NSCR+1))*FUNC12(TPAS*(XSCR+1.),TAB12)
      IF(ITER .EQ. 2) GO TO 37
      SIB=SIB-SCK32
      ITER = 2
      GO TO 33
   37 CONTINUE
      DGF=ATTWGT*DGF1(1)+(1.-ATTWGT)*DGF1(2)
      CDGF=1.-DGF
      XEFFCM=XEFFCM*DGF
   40 CONTINUE
C
C   OUTPUT ATTRITION TO BLUE TASK FORCE
C
      WRITE(6,542)
      WRITE(6,543) (BSITF(KBS),KBS=1,6)
      WRITE(6,544) ( PENK(KBS),KBS=1,6)
      DO 41 KBS=1,NKBS
      BSITF(KBS)=BSITF(KBS)-PENK(KBS)
   41 CONTINUE
      WRITE(6,545) (BSITF(KBS),KBS=1,6)
      IF(IBCP .EQ. 3) WRITE(6,546) CDGF,XEFFCM
C
C   OUTPUT COUNTERKILLS
C
      RSNEW=RSIBAR(IBAR)-SIBCK
      WRITE(6,547) RSIBAR(IBAR),SIBCK,RSNEW
      RSIBAR(IBAR)=RSNEW
```

SUBROUTINE MOVTF

```
C
C  UPDATE SHIPS IN COMMON FOR CTFMOD
C
      XEAAW=BSITF(2)
      XEASWA=BSITF(3)
      XEASWN=BSITF(4)
      XURGS=BSITF(5)
      BSSNDS=BSITF(6)
      LOCTF=DESTF
      WRITE(6, 599)
      WRITE(6,2)
      RETURN
   97 WRITE(6,597)
  597 FORMAT(89H TASK FORCE DIRECTED TO MOVE TO A REGION NOT ADJACENT TO
     1 PREVIOUS REGION.  PROGRAM STOPS. )
      STOP 6404
   98 WRITE(6,598)
  598 FORMAT(72H TASK FORCE DIRECTED TO MOVE TO A REGION EXCEEDING NLOC.
     1  PROGRAM STOPS. )
      STOP 6405
      END
```

SUBROUTINE PLBAB

```
C* DECK PLBAB
      SUBROUTINE PLBAB(L)
C*
C*    PLBAB MODELS THE ATTEMPT BY THE RED AIR ATTACK TO PENETRATE THE
C*    BLUE LAND-BASED AIR BARRIER
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACDB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSDR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
```

SUBROUTINE PLBAB

```
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C*
C* COMDECK COMGA
      COMMON/COMGA/ NTPSLA,BMR(2,3),ESC(2)
C*
C*
      DIMENSION D(2),DA(2),DK(2),DH(2),AA(3),AK(3),AH(3),A(3)
C*
      WRITE(6,1)
      WRITE(6, 101)
    1 FORMAT(51H0--------------------------------------------------)
  101 FORMAT(24H START SUBROUTINE PLBAB )
C*
      TBMR = 0.
      DO 5 I=1,2
      DO 5 K=1,NKRB
    5 TBMR = TBMR + BMR(I,K)
      IF(TBMR .LE. 0.) GO TO 98
C*
      DO 15 KBD=1,NKBDPL
      D(KBD) = 0.
      DO 10 LB=1,NLOC
      D(KBD) = PLFDLL(KBD,LB,L)*PLBLBD(KBD,LB) + D(KBD)
   10 CONTINUE
   15 CONTINUE
      DO 20 KRB=1,NKRB
   20 A(KRB) = BMR(1,KRB) + BMR(2,KRB)
      E = ESC(1) + ESC(2)
      CA = PLCA(L)
C*
      CALL AIRAIR(E,D,A,PLPDED,PLPDDE,PLPDDA,PLPKED,PLPKDE,PLPKDA,
     X            PLPKAD,PLPAJD,PLAEED,PLAEDE,PLAEDA,  CA,1,NKBDPL,NKRB,
     X            IPLAED,IPLADA,EA,EK,EH,DA,DK,DH,AA,AK,AH)
C*
      DO 40 IAB=1,2
      DO 30 KRB=1,NKRB
      IF(A(KRB).LE.0.) GO TO 30
      FACTOR = BMR(IAB,KRB)/A(KRB)
      ATABT(IAB,KRB) = ATABT(IAB,KRB) + AH(KRB)*FACTOR
      BMR(IAB,KRB) = AA(KRB)*FACTOR
   30 CONTINUE
      IF(E.LE.0.) GO TO 40
      FACTOR = ESC(IAB)/E
      AESCAB(IAB) = AESCAB(IAB) + EH*FACTOR
      ESC(IAB) = EA*FACTOR
   40 CONTINUE
      DO 50 KBD=1,NKBDPL
      IF(D(KBD).LE.0.) GO TO 50
      DO 45 LB=1,NLOC
      PLBLBD(KBD,LB) = PLBLBD(KBD,LB) -
     X            (PLBLBD(KBD,LB)*PLFDLL(KBD,LB,L)/D(KBD))*DK(KBD)
   45 CONTINUE
   50 CONTINUE
C*
      DO 60 KRB=1,NKRB
   60 PRINT 110, BMR(1,KRB),BMR(2,KRB),KRB
  110 FORMAT(14H   BMR(1,KRB)=,F10.4,14H   BMR(2,KRB)=,F10.4,10H FOR KRB
```

```
     1 =,I2)
      PRINT 120, ESC(1),ESC(2)
  120 FORMAT(14H       ESC(1)=,F10.4,14H       ESC(2)=,F10.4)
      DO 70 KRB=1,NKRB
   70 PRINT 130, ATABT(1,KRB),ATABT(2,KRB),KRB
  130 FORMAT(14H ATABT(1,KRB)=,F10.4,14H ATABT(2,KRB)=,F10.4,10H FOR KRB
     1 =,I2)
      PRINT 140, AESCAB(1),AESCAB(2)
  140 FORMAT(14H    AESCAB(1)=,F10.4,14H    AESCAB(2)=,F10.4)
      PRINT 150
  150 FORMAT(19H  PLBLBD(KBD,LB) --)
      DO 80 KBD=1,NKBDPL
   80 PRINT 151, (KBD,LB,PLBLBD(KBD,LB),LB=1,NLOC)
  151 FORMAT( 9H  PLBLBD(,I1,1H,,I1,2H)=,F10.4)
C*
      GO TO 99
   98 WRITE(6,198)
  198 FORMAT(44H NO RED AIR ATTACK ON TASK FORCE THIS PERIOD)
   99 WRITE(6,199)
      WRITE(6,2)
  199 FORMAT(25H END OF SUBROUTINE PLBAB )
    2 FORMAT(51H --------------------------------------------------)
C
      RETURN
      END
```

SUBROUTINE POWERP

```
C* DECK POWERP
      SUBROUTINE POWERP(L,ITP)
C*
C*    POWERP CALCULATES POWER PROJECTION RESULTS
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE POWERP

```
C*
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C*
C* COMDECK COMSOR
      COMMON/COMSOR/ FTSORU,ATSORU
C*
C*
C* COMDECK COMOUT
      COMMON/COMOUT/ CWPPAS,CWTPTF,PPSORT,NTPSIM,LTASKF(90)
C*
C*
      DIMENSION AC(2),AS(2),ASA(2),ASH(2),ASK(2),ACK(2)
      DIMENSION SMA(2),SMS(2),SMK(2),AVAILS(2)
      DIMENSION VPKS(4),VPSA(4)
C*
      WRITE(6,1)
      WRITE(6, 201)
    1 FORMAT(51H0---------------------------------------------------)
  201 FORMAT(24H START SUBROUTINE POWERP)
      IF(L .LE. 0) GO TO 96
      WRITE(6,202) L
  202 FORMAT(24H TASK FORCE IS IN REGION,I3,1H.)
C*
C*    BLUE AIRCRAFT FROM CARRIER
C*
      AC(1) = AMIN1(XATTCK,ATTCKI*XEFFCM)
      AC(2) = AMIN1(XFGHTR,FGHTRI*XEFFCM)
      AC(2) = AMAX1(0.,AC(2)-BUCAP*XCAPST)
      P1=PPSORR(1,L)
      P2=PPSORR(2,L)
      IF(P1+P2 .LE. 0.) GO TO 97
      AS(1)=AMAX1(0.,(AC(1)-ATSORU)*P1)
      AS(2)=AMAX1(0.,(AC(2)-FTSORU)*P2)
      TOTSOR = AS(1) + AS(2)
      IF(TOTSOR .LE. 0.) GO TO 98
C*
C*    BLUE ATTACKER/RED SAM INTERACTION
C*
C*    CONVERT MATRICIES TO VECTORS
C*
      DO 20 KRS=1,NPPSAM
      AVAILS(KRS) = PPAVLS(KRS,L)
      DO 20 KBA=1,2
      IND1 = (KBA-1)*NPPSAM + KRS
      IND2 = (KRS-1)*2      + KBA
      VPKS(IND1)=PPPKSA(KRS,KBA)
      VPSA(IND2)=PPPSAS(KBA,KRS)
   20 CONTINUE
C*
      CALL ATRTSS(PPRSAM,PPAVSS,AS,PPPDAS,VPSA,PPPKAS,AVAILS,PPANMS,
     1            PPPDSA,VPKS,PPFASS,PPCAL(L),NPPSAM,2,PPAEGS,PPFASM,
     2            PPFSVS,PPTSCS,IPPAF,IPPAW,SMA,SMS,SMK,ASA,ASH,ASK)
C*
C*    CONVERT SORTIES KILLED TO AIRCRAFT KILLED
C*
```

```
      DO 30 I=1,2
      IF(ASK(I).LE.0) GO TO 23
      IF(PPSORR(I,L).LE.1.) GO TO 22
      ACK(I) = (1.-(1.-ASK(I)/AS(I))**PPSORR(I,L))*AS(I)/PPSORR(I,L)
      GO TO 30
   22 ACK(I) = ASK(I)
      GO TO 30
   23 ACK(I) = 0.
   30 CONTINUE
C*
C*    UPDATE RED SAM AND BLUE AIRCRAFT INVENTORIES AND RECORD SORTIES
C*
      DO 40 KRS=1,NPPSAM
   40 PPRSAM(KRS) = PPRSAM(KRS) - SMK(KRS)
      XATTCK = XATTCK - ACK(1)
      XFGHTR = XFGHTR - ACK(2)
      PPSORT = 0.
      IF(ITP.GT.1) GO TO 50
      DO 45 I=1,2
      PPSORT = PPSORT + ASA(I)
   45 CWPPAS = WFPPAS(I,L)*ASA(I)
      GO TO 60
   50 CONTINUE
      DO 55 I=1,2
      PPSORT = PPSORT + ASA(I)
   55 CWPPAS = CWPPAS + WFPPAS(I,L)*ASA(I)
   60 CONTINUE
C*
C*    PRINT OUT RESULTS
C*
      DO 110 KBA=1,2
      PRINT 210, AS(KBA),AC(KBA),ACK(KBA),KBA
      PRINT 211, ASA(KBA),ASH(KBA),ASK(KBA),KBA
  210 FORMAT(11H   AS(KBA)=,F10.4,11H   AC(KBA)=,F10.4,11H  ACK(KBA)=,F1
     10.4,10H FOR KBA =,I2)
  211 FORMAT(11H  ASA(KBA)=,F10.4,11H  ASH(KBA)=,F10.4,11H  ASK(KBA)=,F1
     10.4,10H FOR KBA =,I2)
  110 CONTINUE
      DO 120 KRS=1,NPPSAM
      PRINT 220, SMA(KRS),SMS(KRS),SMK(KRS),KRS
  220 FORMAT(11H  SMA(KRS)=,F10.4,11H  SMS(KRS)=,F10.4,11H  SMK(KRS)=,F1
     10.4,10H FOR KRS =,I2)
  120 CONTINUE
      PRINT 230, XATTCK,XFGHTR,CWPPAS,PPSORT
  230 FORMAT(11H   XATTCK =,F10.4,11H   XFGHTR =,F10.4,11H   CWPPAS =,F1
     10.4,11H   PPSORT =,F10.4)
C*
      GO TO 99
C
   96 WRITE(6,296)
  296 FORMAT(62H TASK FORCE IS IN REGION ZERO.  NO POWER PROJECTION PERF
     1ORMED.)
      GO TO 99
   97 WRITE(6,297)
  297 FORMAT(88H POWER PROJECTION SORTIE RATES FOR THIS REGION ARE ZERO.
     1   NO POWER PROJECTION PERFORMED.)
      GO TO 99
```

```
   98 WRITE(6,298)
  298 FORMAT(60H NO BLUE AIRCRAFT AVAILABLE.  NO POWER PROJECTION PERFOR
     1MED.)
C
   99 WRITE(6,299)
      WRITE(6,2)
  299 FORMAT(25H END OF SUBROUTINE POWERP)
    2 FORMAT(51H --------------------------------------------------)
C
C*
      RETURN
      END
```

SUBROUTINE PRTRES

```
C* DECK PRTRES
      SUBROUTINE PRTRES(L,ITP)
C*
C*    PRTRES PRINTS OUT THE RESOURCE STATUS --
C*        AT THE START OF COMPAT (ITP = -1),
C*        AFTER DDAY ATTRITION (ITP = 0), AND
C*        AT THE END OF TIME PERIOD ITP (ITP = 1,...,MAXTP).
C*
C*    THIS SUBROUTINE HAS NOT YET BEEN DESIGNED OR PROGRAMMED.
C*
C*
      RETURN
      END
```

SUBROUTINE PRTSUM

```
C* DECK PRTSUM
      SUBROUTINE PRTSUM(LC,ITP)
C*
C*    PRTSUM PRINTS THE SUMMARY PRINTOUT
C*
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE PRTSUM

```
C*
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C*
C* COMDECK COMOUT
      COMMON/COMOUT/ CWPPAS,CWTPTF,PPSORT,NTPSIM,LTASKF(90)
C*
C*
C*    TOTAL SELECTED BLUE RESOURCES BY CATAGORY
C*
      TBSHPC = XEAAW + XEASWA + XEASWN + XURGS
      EFTERM = XEFFCM
      IF(XPLAT.LE.0) EFTERM=TBSHPC
      IF(LC.EQ.0 .AND. ITP.EQ.1) GO TO 7
      IF(LC.EQ.0) GO TO 8
      IF(ITP.EQ.1) GO TO 6
      CWTPTF = CWTPTF + EFTERM*WFTFL(LC)
      GO TO 8
    6 CWTPTF = EFTERM*WFTFL(LC)
      GO TO 8
    7 CWTPTF = 0.
    8 CONTINUE
      TBLBDS = 0.
      DO 10 KBD=1,NKBDPL
      DO 10 L=1,NLOC
   10 TBLBDS = TBLBDS + PLBLBD(KBD,L)
      TBSUBS = 0.
      DO 20 L=1,NLOC
   20 TBSUBS = TBSUBS + BSIBAR(L)
      TBSUBS = TBSUBS + BSSNDS
C*
C*    TOTAL SELECTED RED RESOURCES BY CATAGORY
C*
      TRSHIP = 0.
      IF(NKRS.LT.3) GO TO 40
      NKRSS = NKRS - 2
      DO 30 L=1,NLOC
      DO 30 KRSS=1,NKRSS
      KRS=KRSS+2
   30 TRSHIP=TRSHIP+RS(KRS,L)
   40 CONTINUE
      TRSUBS = 0.
      IF(NKRS-1) 70,60,50
   50 DO 55 L=1,NLOC
   55 TRSUBS = TRSUBS + RS(1,L) + RS(2,L) + RSIBAR(L)
      GO TO 80
   60 DO 65 L=1,NLOC
   65 TRSUBS = TRSUBS + RS(1,L) + RSIBAR(L)
      GO TO 80
   70 DO 75 L=1,NLOC
   75 TRSUBS = TRSUBS + RSIBAR(L)
   80 CONTINUE
      TRESCS = 0.
      TRBMRS = 0.
      DO 90 IAB=1,2
      DO 85 KRB=1,NKRB
```

SUBROUTINE PRTSUM

```
   85 TRBMRS = TRBMRS + ATABT(IAB,KRB)
      TRESCS = TRESCS + AESCAB(IAB)
   90 CONTINUE
C*
C*    RECORD SUMMARY OUTPUT FOR CURRENT ITP
C*
      MST=10
      WRITE(MST,1030) ITP,XEFFCM,CWTPTF,TBSHPC,TBSUBS,XFGHTR,XATTCK,
     1TBLBDS,PPSORT,CWPPAS,TRSHIP,TPSUBS,TRBMRS,TRESCS,AINTCT
 1030 FORMAT(1H ,I3,2F8.4,2F8.2,5F9.2,4X,2F8.2,3F9.2)
C*
      PPSORT = 0.
C*
      RETURN
      END
```

SUBROUTINE SHPSHP

```
C* DECK SHPSHP
      SUBROUTINE SHPSHP(L,ITP)
C*
C*    SHPSHP MODELS SURFACE SHIP VS SURFACE SHIP WARFARE
C*
C
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACDB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPCAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
```

SUBROUTINE SHPSHP

```
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C
C* COMDECK COMCTF
      COMMON /COMCTF/ XEFFCM,FGHTRI,ATTCKI,XCAPST
C*
C
C* COMDECK COMSOR
      COMMON/COMSOR/ FTSORU,ATSORU
C*
C
      DIMENSION BSS(5),BSSKO(5,2),BSSK(5),BSNEW(5),RSSV(8),RSSKAC(8)
      DIMENSION RSS(8),RSSKO(8,2),RSSK(8),RSNEW(8)
      WRITE(6,1000) ITP,L
 1000 FORMAT(25HOREMINDER--THIS IS PERIOD ,I4,25H, TASK FORCE IS IN REGI
     1ON,I3,1H.)
      WRITE(6,1)
      WRITE(6, 501)
    1 FORMAT(51H0--------------------------------------------------)
  501 FORMAT(24H START SUBROUTINE SHPSHP)
      IF(NKRS .LT. 3) GO TO 98
C
C  BLUE SHIPS IN TASK FORCE
C
      BSS(1)=XPLAT
      BSS(2)=XEAAW
      BSS(3)=XEASWA
      BSS(4)=XEASWN
      BSS(5)=XURGS
      DO 6 KBSS=1,5
      BSSKO(KBSS,1)=0.
      BSSKO(KBSS,2)=0.
      BSSK(KBSS)=0.
    6 CONTINUE
C
C  RED SURFACE SHIPS VULNERABLE TO ATTACK
C
      NKRSS=NKRS-2
      TRSSV=0.
      DO 3 KRSS=1,NKRSS
      KRS=KRSS+2
      RSSV(KRSS)=RS(KRS,L)*SSFRSV(KRSS,L)
      RSSKAC(KRSS)=0.
      RSSKO(KRSS,1)=0.
      RSSKO(KRSS,2)=0.
      RSSK(KRSS)=0.
      TRSSV=TRSSV+RSSV(KRSS)
    3 CONTINUE
      IF(TRSSV .GT. 0.) GO TO 4
      WRITE(6,504)
  504 FORMAT(72H NO VULNERABLE RED SURFACE SHIPS IN REGION, HENCE NO COM
     1BAT TAKES PLACE.   )
      WRITE(6,561)
      WRITE(6,565) (BSS(I),I=1,5)
      WRITE(6,567) XEFFCM
      GO TO 99
    4 CONTINUE
      ISS=0
```

SUBROUTINE SHPSHP

```
      IBAC=0
      FRK=0.
      IF(XPLAT .EQ. 0.) GO TO 10
C
C  BLUE AIRCRAFT FROM CARRIER
C
      FA=AMIN1(FGHTRI*XEFFCM,XFGHTR-FTSORU)
      FA=AMAX1(0.0,FA)
      FA=AMAX1(FA-BUCAP*XCAPST,0.)
      AA=AMIN1(ATTCKI*XEFFCM,XATTCK-ATSORU)
      AA=AMAX1(0.0,AA)
      BACA=SSCFA*FA+AA
      BACR=0.
      DO 5 KRSS=1,NKRSS
      BACR=BACR+SSBACR(KRSS)*RSSV(KRSS)
    5 CONTINUE
      WRITE(6,505) BACA,BACR
  505 FORMAT(29H BLUE AIRCRAFT FROM CARRIER--,F8.2,63H.  BLUE AIRCRAFT R
     1EQUIRED TO DESTROY ALL VULNERABLE RED SHIPS--,F8.2,1H.)
      FAK = 0.
      AAK = 0.
      IF(BACR.LE.0.0) GO TO 150
      IF(AA.LT.BACR .AND. SSCFA.GT.0.) GO TO 120
      AAF = AMIN1(AA,BACR)
      DO 110 KRSS=1,NKRSS
  110 AAK = AAK + SSBACR(KRSS)*RSSV(KRSS)*SSFBAK(1,KRSS)*AAF/BACR
      ATSORU = ATSORU + (AAF-AAK)*PAFCNF
      GO TO 140
  120 CONTINUE
      AAF = AMIN1(AA,BACR)
      FAF = (BACR-AAF)/SSCFA
      FAF = AMIN1(FA,FAF)
      DO 130 KRSS=1,NKRSS
      AAK = AAK + SSBACR(KRSS)*RSSV(KRSS)*SSFBAK(1,KRSS)*AAF/BACR
      FAK = FAK + SSBACR(KRSS)*RSSV(KRSS)*SSFBAK(2,KRSS)*FAF/BACR
  130 CONTINUE
      ATSORU = ATSORU + (AAF-AAK)*PAFCNF
      FTSORU = FTSORU + (FAF-FAK)*PFFCNF
  140 CONTINUE
      XATTCK = XATTCK - AAK
      XFGHTR = XFGHTR - FAK
  150 CONTINUE
      WRITE(6,510) XATTCK,AAK,ATSORU
      WRITE(6,511) XFGHTR,FAK,FTSORU
  510 FORMAT(9H  XATTCK=,F10.4,7H,  AAK=,F10.4,10H,  ATSORU=,F10.4)
  511 FORMAT(9H  XFGHTR=,F10.4,7H,  FAK=,F10.4,10H,  FTSORU=,F10.4)
      IF(BACA .LT. BACR) GO TO 7
      WRITE(6,507)
  507 FORMAT(115H SINCE THERE ARE SUFFICIENT BLUE AIRCRAFT TO DESTROY AL
     1L VULNERABLE RED SHIPS, THERE IS NO ATTRITION TO BLUE SHIPS./1H ,
     251H(THE SHIP-TO-SHIP INTERACTION DOES NOT TAKE PLACE.))
      IBAC=2
      GO TO 50
    7 IBAC=1
      FRK=BACA/BACR
C
   10 CONTINUE
```

```
      TRSS=0.
      DO 8 KRSS=1,NKRSS
      RSSKAC(KRSS)=RSSV(KRSS)*FRK
      RSS(KRSS)=RSSV(KRSS)-RSSKAC(KRSS)
      TRSS=TRSS+RSS(KRSS)
    8 CONTINUE
C
C   SURFACE SHIP VS SURFACE SHIP INTERACTION
C
      ISS=1
      TBSSNC=XEAAW+XEASWA+XEASWN+XURGS
      TBSS=TBSSNC+XPLAT
      TBA=TBSSNC
      TBSSR=TBSSNC
      TRSSR=TRSS
      ITER=1
C   ATTRITION TO RED
   11 IF(TBA .LE. 0.) GO TO 20
      IF(TRSS .LE. 0.) GO TO 20
      IF(ISSBR .EQ. 1) GO TO 15
      TERM=1.-(1.-SSPBDR)**TRSS
      TERM=TERM*SSPBKR/AMAX1(1.,TRSS)
      TERM=(1.-TERM)**TBA
      FRK=1.-TERM
      GO TO 17
   15 FRK=SSPBDR*(1.-(1.-SSPBKR/AMAX1(1.,SSPBDR*TRSS))**TBA)
   17 DO 19 KRSS=1,NKRSS
      RSSKO(KRSS,ITER)=RSS(KRSS)*FRK
      TRSSR=TRSSR-RSSKO(KRSS,ITER)
   19 CONTINUE
   20 IF(ITER .EQ. 2) GO TO 34
      TRA=TRSSR
C   ATTRITION TO BLUE
   21 IF(TRA .LE. 0.) GO TO 30
      IF(TBSS .LE. 0.) GO TO 30
      TBSSR=TBSSNC
      IF(ISSRB .EQ. 1) GO TO 25
      TERM=1.-(1.-SSPRDB)**TBSS
      TERM=TERM/AMAX1(1.,TBSS)
      TC=1.-TERM*SSPRKC
      TS=1.-TERM*SSPRKB
      FBCK=1.-TC**TRA
      FBSK=1.-TS**TRA
      GO TO 27
   25 TERM=SSPRDB/AMAX1(1.,TBSS)
      FBCK=1.-(1.-SSPRKC*TERM)**TRA
      FBSK=1.-(1.-SSPRKB*TERM)**TRA
   27 DO 29 KBSS=2,5
      BSSKO(KBSS,ITER)=BSS(KBSS)*FBSK
      TBSSR=TBSSR-BSSKO(KBSS,ITER)
   29 CONTINUE
      BSSKO(1,ITER)=XPLAT*FBCK
   30 IF(ITER .EQ. 2) GO TO 32
C   BRANCHING LOGIC FOR SHOOT-AND-SHOOT-BACK ATTRITION SCHEME
      ITER=2
      TRA=TRSS
      GO TO 21
```

```
   32 TBA=TBSSR
      GO TO 11
   34 CONTINUE
      OMEGA1=TBSS/(TBSS+TRSS)
      OMEGA2=1.-OMEGA1
      DO 40 KBSS=1,5
      BSSK(KBSS)=OMEGA1*BSSKO(KBSS,1)+OMEGA2*BSSKO(KBSS,2)
      BSNEW(KBSS)=BSS(KBSS)-BSSK(KBSS)
   40 CONTINUE
      DEGF=0.
      IF(XPLAT .GT. 0. ) DEGF=BSSK(1)/XPLAT
      XEFFCM=XEFFCM*(1.-DEGF)
      BSNEW(1)=XPLAT
      BSSK(1)=0.
      DO 45 KRSS=1,NKRSS
      RSSK(KRSS)=OMEGA1*RSSKO(KRSS,1)+OMEGA2*RSSKO(KRSS,2)
      KRS=KRSS+2
      RSNEW(KRSS)=RS(KRS,L)-RSSKAC(KRSS)-RSSK(KRSS)
   45 CONTINUE
      IF(IBAC.EQ.0) GO TO 60
C   COMPUTE ATTRITION TO AIRCRAFT ON CARRIERS
      ENACD = 0.
      PIACD = 0.
      FACD  = 0.
      FDMCV = 0.
      ADMCV = 0.
      TIACFT = ATTCKI + FGHTRI
      IF(TIACFT.LE.0.) GO TO 47
      IF(FBCK.LE.0.) GO TO 47
      ATT = BSSK(1)/FBCK
      DO 46 KRS=3,NKRS
      KRSS=KRS-2
      FRAC=RSS(KRSS)/TRSS
      ENACD = ENACD + ENACDS(KRS)*FRAC
   46 CONTINUE
      PIACD = ENACD/TIACFT
      PIACD = AMIN1(0.9999,PIACD)
      FACD = 1.-(1.-PIACD)**ATT
      FDMCV = XFGHTR*FACD
      ADMCV = XATTCK*FACD
      XFGHTR = XFGHTR - FDMCV
      XATTCK = XATTCK - ADMCV
   47 CONTINUE
      GO TO 60
C
C   BLUE AIRCRAFT DESTROY ALL VULNERABLE RED SHIPS
C
   50 CONTINUE
      DO 51 KRSS=1,NKRSS
      RSSKAC(KRSS)=RSSV(KRSS)
      KRS=KRSS+2
      RSNEW(KRSS)=RS(KRS,L)-RSSKAC(KRSS)
   51 CONTINUE
C
C   OUTPUT RESULTS
C
   60 CONTINUE
```

```
      IF(IBAC .NE. 2) GO TO 61
      WRITE(6,561)
      WRITE(6,565) (BSS(I),I=1,5)
      WRITE(6,567) XEFFCM
      GO TO 70
   61 WRITE(6,560)
      WRITE(6,561)
      WRITE(6,562) (  BSS(KBSS),KBSS=1,5)
      WRITE(6,563) ( BSSK(KBSS),KBSS=1,5)
      WRITE(6,564) (BSNEW(KBSS),KBSS=1,5)
      WRITE(6,566) DEGF,XEFFCM
  560 FORMAT(26H BLUE SURFACE SHIP RESULTS)
  561 FORMAT(1H ,4X,17HKIND OF BLUE SHIP,21X,33HXPLAT  XEAAW XEASWA XEAS
     1WN  XURGS )
  562 FORMAT(1H ,4X,32HINITIAL BLUE SHIPS IN TASK FORCE,4X,5F7.2)
  563 FORMAT(1H ,4X,20HBLUE SHIPS DESTROYED,16X,5F7.2)
  564 FORMAT(1H ,4X,34HRESULTANT BLUE SHIPS IN TASK FORCE,2X,5F7.2)
  565 FORMAT(1H ,4X,24HBLUE SHIPS IN TASK FORCE,12X,5F7.2)
  566 FORMAT(1H ,4X,30HCARRIER CAPABILITY DEGRADED BY,F7.4,49H, NEW RELA
     1TIVE CARRIER CAPABILITY (XEFFCM) EQUALS,F7.4,1H.)
  567 FORMAT(1H ,4X,43HRELATIVE CARRIER CAPABILITY (XEFFCM) EQUALS,F7.4,
     11H.)
   70 WRITE(6,570)
      WRITE(6,571) (I,I=3,NKRS)
      WRITE(6,572) (  RS(KRS,L),KRS=3,NKRS)
      WRITE(6,573) (RSSV(KRSS),KRSS=1,NKRSS)
      IF(IBAC .GT. 0) WRITE(6,574) (RSSKAC(KRSS),KRSS=1,NKRSS)
      IF(ISS .EQ. 1) WRITE(6,575) (RSSK(KRSS),KRSS=1,NKRSS)
      WRITE(6,576) (RSNEW(KRSS),KRSS=1,NKRSS)
  570 FORMAT(118H RED SURFACE SHIP RESULTS (NOTE--RED SHIP KINDS 1 AND 2
     1 ARE SUBMARINES, WHICH DO NOT PARTICIPATE IN THIS INTERACTION.)   )
  571 FORMAT(1H ,4X,16HKIND OF RED SHIP,20X,8I7)
  572 FORMAT(1H ,4X,27HINITIAL RED SHIPS IN REGION,9X,8F7.2)
  573 FORMAT(1H ,4X,36HRED SHIPS VULNERABLE TO BLUE ATTACK ,8F7.2)
  574 FORMAT(1H ,4X,36HRED SHIPS DESTROYED BY BLUE AIRCRAFT,8F7.2)
  575 FORMAT(1H ,4X,36HRED SHIPS DESTROYED BY BLUE SHIPS    ,8F7.2)
  576 FORMAT(1H ,4X,29HRESULTANT RED SHIPS IN REGION,7X,8F7.2)
      IF(IBAC.NE.1) GO TO 75
      PRINT 284, ENACD,PIACD,FACD
      PRINT 285, FDMCV,XFGHTR
      PRINT 286, ADMCV,XATTCK
  284 FORMAT (8H  ENACD=,F10.4,8H  PIACD=,F10.4,8H   FACD=,F10.4)
  285 FORMAT (8H  FDMCV=,F10.4,8H XFGHTR=,F10.4)
  286 FORMAT (8H  ADMCV=,F10.4,8H XATTCK=,F10.4)
   75 CONTINUE
C
C  UPDATE APPROPRIATE QUANTITIES
C
      DO 81 KRSS=1,NKRSS
      KRS=KRSS+2
      RS(KRS,L)=RSNEW(KRSS)
   81 CONTINUE
      IF(IBAC .EQ. 2) GO TO 85
      XEAAW= BSNEW(2)
      XEASWA=BSNEW(3)
      XEASWN=BSNEW(4)
      XURGS= BSNEW(5)
```

```
   85 CONTINUE
      GO TO 99
   98 WRITE(6,598) NKRS
  598 FORMAT(6H NKRS=,I2,113H CURRENTLY.  NKRS MUST BE AT LEAST 3 (I.E.,
     1 THERE MUST BE AT LEAST ONE KIND OF RED SURFACE SHIP).  PROGRAM ST
     2OPS.)
      STOP 6407
   99 WRITE(6,599)
      WRITE(6,2)
  599 FORMAT(25H END OF SUBROUTINE SHPSHP)
    2 FORMAT(51H --------------------------------------------------)
C
      RETURN
      END
```

SUBROUTINE SUBSUB

```
C* DECK SUBSUB
      SUBROUTINE SUBSUB(L)
C*
C*    SUBSUB MODELS BLUE SUBMARINES IN DIRECT SUPPORT OF THE TASK FORCE
C*    VERSUS RED SUBMARINES AND RED SURFACE SHIPS IN THE SAME LOCATION
C*
C
C* COMDECK COMINP
      COMMON   NEPD(1)
      COMMON   AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON   AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON   AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON   ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON   ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON   ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON   AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON   AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON   BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON   BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON   CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON   CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON   DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON   DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON   ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON   FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON   FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON   HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON   IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON   IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON   IPPAF,IPPAW
      COMMON   LGTHMP(6),LTFMP(6)
      COMMON   MAXTP,MIMP
      COMMON   NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON   PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON   PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON   PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON   PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON   PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON   PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON   PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON   PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON   PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON   PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON   RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON   RS(10,5),RSIBAR(5)
      COMMON   SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON   SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON   SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON   SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON   SSFBAK(2,8),SSPRKC
      COMMON   TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON   THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON   UBAEW,UBAEWL,UBASW,UBASWL
      COMMON   VBT(3),VCAP,VI
      COMMON   WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON   WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON   XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
```

SUBROUTINE SUBSUB

```
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C
      DIMENSION RSKK(10),RSNEW(10)
C
      WRITE(6,1)
      WRITE(6, 501)
    1 FORMAT(51H0--------------------------------------------------)
  501 FORMAT(24H START SUBROUTINE SUBSUB)
      IF(NKRS .LT. 3) GO TO 98
C
C  SET UP NUMBERS OF COMBATANTS
C
      WRITE(6,505) L
  505 FORMAT(24H TASK FORCE IS IN REGION ,I3)
      IF(BSSNDS .LE. 0.) GO TO 97
      BSA=BSSNDS
      BSCS=BSA*SBFBCS
      RST=RS(1,L)+RS(2,L)
      RFT=0.
      DO 5 KRS=3,NKRS
      RFT=RFT+RS(KRS,L)
    5 CONTINUE
      IF(RST+RFT .LE. 0.) GO TO 96
      RSA=RST*SBFRSA(L)
      RSC=RSA*SBFRSC
      DO 6 KRS=1,NKRS
      RSKK(KRS)=0.
      RSNEW(KRS)=0.
    6 CONTINUE
C
C  COMPUTE ATTRITION--RED SUB/BLUE SUB BATTLE
C
      CALL BINOAT(BSCS,RSA ,SBPBDS,SBPBKS,1.,1,1,RSK1)
      RS1=SBFRSC*(RSA-RSK1)
      CALL BINOAT(RS1 ,BSA ,SBPSDB,SBPSKB,1.,1,1,BSK1)
      CALL BINOAT(RSC ,BSA ,SBPSDB,SBPSKB,1.,1,1, BSK2)
      BS2=SBFBCS*(BSA-BSK2)
      CALL BINOAT(BS2 ,RSA ,SBPBDS,SBPBKS,1.,1,1,RSK2)
      DENOM=BSCS+RSC
      IF(DENOM .EQ. 0.) GO TO 10
      BSK=(BSK1*BSCS + BSK2*RSC)/DENOM
      RSK=(RSK1*BSCS + RSK2*RSC)/DENOM
      GO TO 11
   10 BSK=0.
      RSK=0.
   11 CONTINUE
      RSL=RST-RSK
C
C  COMPUTE ATTRITION--RED SURFACE SHIP/BLUE SUB BATTLE
C
      RFA=RFT*SBFRFA(L)
      BSAF=BSA-BSK
      BSCF=BSAF*SBFBCF
      CALL BINOAT(BSCF,RFA ,SBPBDF,SBPBKF,1.,1,1,RFK)
      RF=SBFRFC*(RFA-RFK)
      CALL BINOAT(RF  ,BSAF,SBPFDB,SBPFKB,1.,1,1,BSKF)
```

```
      RFL=RFT-RFK
      TBSK=BSK+BSKF
      BSSNEW=BSSNDS-TBSK
      IF(RST .EQ. 0.) GO TO 15
      DO 14 KRS=1,2
      RSKK(KRS)=RSK*RS(KRS,L)/RST
      RSNEW(KRS)=RS(KRS,L)-RSKK(KRS)
   14 CONTINUE
   15 IF(RFT .EQ. 0.) GO TO 17
      DO 16 KRS=3,NKRS
      RSKK(KRS)=RFK*RS(KRS,L)/RFT
      RSNEW(KRS)=RS(KRS,L)-RSKK(KRS)
   16 CONTINUE
   17 CONTINUE
C
C   OUTPUT RESULTS
C
      IF(RST .GT. 0.) GO TO 20
      WRITE(6,545)
  545 FORMAT(67H NO RED SUBMARINES IN REGION--SUBMARINE BATTLE DOES NOT
     1TAKE PLACE.)
      WRITE(6,531)
      WRITE(6,521) BSSNDS
      WRITE(6,534)  RFT
      WRITE(6,523)
      WRITE(6,536) RFA
      GO TO 35
   20 CONTINUE
      WRITE(6,520)
      WRITE(6,521) BSSNDS
      WRITE(6,522)RST
      WRITE(6,523)
      WRITE(6,524) RSA
      WRITE(6,525) RSC
      WRITE(6,526) BSCS
      WRITE(6,527)  BSK
      WRITE(6,528)RSK
      WRITE(6,529) BSAF
      WRITE(6,530) RSL
  520 FORMAT(56H RESULTS OF THE BLUE SUBMARINE/RED SUBMARINE INTERACTION
     1)
  521 FORMAT(47H      INITIAL BLUE SUBMARINES IN TASK FORCE----- ,  F7.2)
  522 FORMAT(1H+,63X,42HINITIAL RED SUBMARINES IN REGION-----------,F7.2)
  523 FORMAT(47H       (ALL BLUE SUBS ENGAGE IN COMBAT.)        ,F7.2)
  524 FORMAT(1H+,63X,42HRED SUBS ENGAGING IN COMBAT---------------,F7.2)
  525 FORMAT(47H      RED SUBMARINES CAPABLE OF ATTACKING BLUE-- ,F7.2)
  526 FORMAT(1H+,63X,42HBLUE SUBMARINES CAPABLE OF ATTACKING RED--,F7.2)
  527 FORMAT(47H      BLUE SUBMARINES KILLED BY RED SUBMARINES-- ,F7.2)
  528 FORMAT(1H+,63X,42HRED SUBMARINES KILLED BY BLUE SUBMARINES--,F7.2)
  529 FORMAT(47H      RESULTANT BLUE SUBMARINES IN TASK FORCE--- ,F7.2)
  530 FORMAT(1H+,63X,42HRESULTANT RED SUBMARINES IN REGION--------,F7.2)
      IF(RFT .LE. 0) GO TO 45
      WRITE(6,531)
      WRITE(6,533) BSAF
      WRITE(6,534) RFT
      WRITE(6,535)
      WRITE(6,536) RFA
```

```
 35 WRITE(6,537) RF
    WRITE(6,538) BSCF
    WRITE(6,539) BSKF
    WRITE(6,540) RFK
    WRITE(6,529) BSSNEW
    WRITE(6,542) RFL
531 FORMAT(59H RESULTS OF THE BLUE SUBMARINE/RED SURFACE SHIP INTERACT
   1ION)
533 FORMAT(47H      BLUE SUBS ATTACKING RED SURFACE SHIPS-----,F7.2)
534 FORMAT(1H+,63X,42HINITIAL RED SURFACE SHIPS IN REGION-------,F7.2)
535 FORMAT(47H       (ALL BLUE SUBS THAT SURVIVED RED SUBS)      )
536 FORMAT(1H+,63X,42HRED SURFACE SHIPS ENGAGING IN COMBAT------,F7.2)
537 FORMAT(47H      RED SURF.SHIPS CAPABLE OF ATTACKING BLUE-- ,F7.2)
538 FORMAT(1H+,63X,42HBLUE SUBS CAPABLE OF ATTACKING RED SURF.--,F7.2)
539 FORMAT(47H      BLUE SUBS KILLED BY RED SURFACE SHIPS----- ,F7.2)
540 FORMAT(1H+,63X,42HRED SURFACE SHIPS KILLED------------------,F7.2)
542 FORMAT(1H+,63X,42HRESULTANT RED SURFACE SHIPS IN REGION-----,F7.2)
    GO TO 50
 45 WRITE(6,550)
550 FORMAT(86H NO RED SURFACE SHIPS IN REGION--BLUE SUB/RED SURFACE SH
   1IP BATTLE DOES NOT TAKE PLACE.)
 50 WRITE(6,560) BSSNDS,TBSK,BSSNEW
    WRITE(6,570)
    WRITE(6,571) ( KRS,KRS=1,NKRS)
    WRITE(6,572) (RS(KRS,L),KRS=1,NKRS)
    WRITE(6,573) (RSKK(KRS),KRS=1,NKRS)
    WRITE(6,574) (RSNEW(KRS),KRS=1,NKRS)
560 FORMAT(23H OVERALL BLUE RESULTS--,F7.2,28H BLUE SSN(DS) INITIALLY
   1LESS,F7.2,14H KILLED YIELDS,F7.2,11H SURVIVING.)
570 FORMAT(72H OVERALL RED RESULTS, BY KIND OF RED SHIP.  (ATTRITION I
   1S PROPORTIONAL.) )
571 FORMAT(1H ,4X,16HKIND OF RED SHIP,13X,10I7)
572 FORMAT(1H ,4X,27HINITIAL RED SHIPS IN REGION, 2X,10F7.2)
573 FORMAT(1H ,4X,16HRED SHIPS KILLED,13X,10F7.2)
574 FORMAT(1H ,4X,29HRESULTANT RED SHIPS IN REGION,10F7.2)
    BSSNDS=BSSNEW
    DO 80 KRS=1,NKRS
    RS(KRS,L) =RSNEW(KRS)
 80 CONTINUE
    GO TO 99
 96 WRITE(6,596)
596 FORMAT(69H NO RED SUBMARINES OR SURFACE SHIPS IN REGION--NO COMBAT
   1 TAKES PLACE.)
    GO TO 99
 97 WRITE(6,597)
597 FORMAT(57H NO BLUE SUBMARINES IN TASK FORCE--NO COMBAT TAKES PLACE
   1.)
    GO TO 99
 98 WRITE(6,598) NKRS
598 FORMAT(6H NKRS=,I2,113H CURRENTLY.  NKRS MUST BE AT LEAST 3 (I.E.,
   1 THERE MUST BE AT LEAST ONE KIND OF RED SURFACE SHIP).  PROGRAM ST
   2OPS. )
    STOP 6410
 99 WRITE(6,599)
    WRITE(6,2)
599 FORMAT(25H END OF SUBROUTINE SUBSUB)
  2 FORMAT(51H --------------------------------------------------)
C
C*
    RETURN
    END
```

SUBROUTINE TIMET

```
C* DECK TIMET
      SUBROUTINE TIMET(ICYCLE)
C
C       THEATER CONTROL ROUTINE FOR I/O
C*
C* COMDECK COMINP
      COMMON   NEPD(1)
      COMMON   AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON   AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON   AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON   ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON   ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON   ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON   AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON   AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON   BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON   BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON   CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON   CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON   DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON   DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON   ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON   FAACA(5),FFACA(5),FFACE(5),FACOB(5,2),FHSK(2)
      COMMON   FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON   HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON   IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON   IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON   IPPAF,IPPAW
      COMMON   LGTHMP(6),LTFMP(6)
      COMMON   MAXTP,MIMP
      COMMON   NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON   PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON   PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON   PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON   PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON   PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON   PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON   PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON   PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON   PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON   PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON   RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON   RS(10,5),RSIBAR(5)
      COMMON   SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON   SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON   SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON   SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON   SSFBAK(2,8),SSPRKC
      COMMON   TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON   THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON   UBAEW,UBAEWL,UBASW,UBASWL
      COMMON   VBT(3),VCAP,VI
      COMMON   WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON   WRLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON   XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON   XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON   ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
```

SUBROUTINE TIMET

```
C*
C* COMDECK COMIGO
      COMMON/COMIGO/ IGO
C*
C*
      DIMENSION IREC(21),ZREC(21)        ,ZEPD(1),IVAL(15),ZVAL(15)
      EQUIVALENCE (IREC(1),ZREC(1)),(IDN,IREC(1)),(IFN,IREC(2)),
     1 (KBA,IREC(8)),(KREG,IREC(10)),(KSA,IREC(9)),(IT,IREC(7)),
     2 (IREC(4),IID),(ICODE,IREC(3)),(IST,IREC(4)),(ITO,IREC(5)),
     3 (IINC,IREC(6)),(IVAL,IREC(7)),(IVAL(1),ZVAL(1)),(NEPD(1),CEPD,
     4 ZEPD(1))
C*
   87 READ(15)IREC
      IF(EOF(15))90,91
   91 IF(IDN.NE.9999)GOTO92
      IGO=IFN
   89 IF (IGO-ICYCLE) 81,87,81
   90 IGO=99999
      GOTO81
C     PROCESS THE RECORD
C    INTEGER VALUES ARE TO BE INCREMENTED OR REPLACED IF NFN=1--REAL
C     VALUES, IF NFN=2
   92 KK=0
      ASSIGN 103 TO MGO
      IF(IFN.NE.1)GOTO94
      IF(ICODE.EQ.0)ASSIGN 104 TO MGO
      GOTO 97
   94 ASSIGN 105 TO MGO
      IF(ICODE.EQ.0)ASSIGN 106 TO MGO
   97 DO 98 K=IST,ITO,IINC
      KK=KK+1
      GO TO MGO,(103,104,105,106)
  103 NEPD(K)=NEPD(K)+IVAL(KK)
      GOTO98
  104 NEPD(K)=IVAL(KK)
      GOTO98
  105 ZEPD(K)=ZEPD(K)+ZVAL(KK)
      GOTO98
  106 ZEPD(K)=ZVAL(KK)
   98 CONTINUE
      GOTO87
   81 CONTINUE
      RETURN
      END
```

PRCGRAM INP

```
C* DECK INP
      OVERLAY(OVER,1,0)
      PROGRAM INP
C
C* COMDECK COMINP
      COMMON  NEPD(1)
      COMMON  AAAEDA(2),AAAEDE(2),AAAEED(1),AACA,AAPAJO(2),AAPDDA(2)
      COMMON  AAPDDE(2),AAPDED(1),AAPKAD(2,2),AAPKDA(2,2),AAPKDE(2,1)
      COMMON  AAPKED(1,2),AASRAA(5),AASRED,AASRFA(5),AASRFE(5),AASRID
      COMMON  ABANM(2),ABAVLS(2),ABCAS,ABESGS(2),ABFASS(2)
      COMMON  ABFSM(2),ABFVS(2),ABPDA(2),ABPKA(2),ABPSA(2,2),ABPDS(2)
      COMMON  ABPKS(2,2),ABTSC(2),ABVGSS(2),ABRSAM(2)
      COMMON  AEWD,AESCAB(2),ASWF,ATABT(2,3),ATTWGT,AVAILE(5,2)
      COMMON  AINTCT,AVAILT(5,2,3),AVALED(5,2),AWRCBB
      COMMON  BACCDW(6),BACPCK(6),BAREAQ(5),BARELQ(5),BARLQ(5),BMTMIN(5)
      COMMON  BARLTH(5),BECDW(6),BEDW(10),BSIBAR(5),BSSNDS,BUCAP
      COMMON  CACDWO,CAPMLQ(5),CAPMQ(5),CAPMR,CAPSTQ(5)
      COMMON  CPAGV,CPBPK(6),CPBSCK(10),CPRPK(10),CPRSCK(6),CSCDWO
      COMMON  DDFAC(10),DDPKC(10),DDPKS(10),DDRKAA(10),DDRKBA(10)
      COMMON  DDRSA(10),DDSPA(10),DLIA,D1T(2,3),D2T(2,3)
      COMMON  ESLR,ESRQ(5),ENACDT(4),ENACDS(10)
      COMMON  FAACA(5),FFACA(5),FFACE(5),FACDB(5,2),FHSK(2)
      COMMON  FM3(6),FPPL1,FPPL2,FSTAQ(5),FSTGAQ(5)
      COMMON  HRMAAW,HRMASW,HRMURG,HRTAAW,HRTASW,HRTURG
      COMMON  IAADA,IAAED,IABAF,IABAW,IABAEQ,IATKRT(5),IATRIA,ICTL(5)
      COMMON  IDDAC,IDDAS,IKRAS(5),IPLADA,IPLAED,IRSUBA(5),ISSBR,ISSRB
      COMMON  IPPAF,IPPAW
      COMMON  LGTHMP(6),LTFMP(6)
      COMMON  MAXTP,MIMP
      COMMON  NABSAM,NKRB,NKRS,NKBDPL,NLOC,NPPSAM
      COMMON  PARK,PASS(2),PBDRN(2),PBDRS(2),PBKRN(2),PBKRS(2)
      COMMON  PDIN,PKAT1,PKDF1,PKASW,PKIIN,PKIN,PKPLDT(4),PKPL1,PKPL2
      COMMON  PKSST(4),PRSM(10,5,6),PRWLNQ(5)
      COMMON  PLAEDA(2),PLAEDE(2),PLAEED,PLBLBD(2,5),PLCA(5)
      COMMON  PLFDLL(5,5,2),PLPAJO(3),PLPDDA(2),PLPDDE(2),PLPDED
      COMMON  PLPKAD(3,2),PLPKDA(2,3),PLPKDE(2),PLPKED(2)
      COMMON  PAFCNF,PFFCNF,PPSORR(2,5),PPPSAS(2,2),PPPKSA(2,2)
      COMMON  PPRSAM(2),PPAVSS(2),PPPKAS(2),PPAVLS(2,5),PPANMS(2)
      COMMON  PPPDSA(2),PPFSVS(2),PPTSCS(2),PPCAL(5)
      COMMON  PPPDAS(2),PPFASS(2),PPAEGS(2),PPFASM(2)
      COMMON  RACCDW(10),RACPCK(10),RECDW(10),REDW(6),RARBAB(3)
      COMMON  RS(10,5),RSIBAR(5)
      COMMON  SBFBCF,SBFBCS,SBFRFA(5),SBFRFC,SBFRSA(5),SBFRSC
      COMMON  SBPBDF,SBPBDS,SBPBKF,SBPBKS,SBPFDB,SBPFKB,SBPSDB,SBPSKB
      COMMON  SMALLR,SSDAAW,SSDASW,SSDURG,STARQ(5),STSALV,SUBSOR,SHEL
      COMMON  SSBACR(8),SSCFA,SSFRSV(8,5),SSPBDR,SSPBKR,SSPRDB,SSPRKB
      COMMON  SSFBAK(2,8),SSPRKC
      COMMON  TAB10T(20,4),TAB12(20),TAB13T(20,4),TCAP,THSCAQ(5)
      COMMON  THSCTQ(5),TPAS,TPS,T1,T2,T3,T4
      COMMON  UBAEW,UBAEWL,UBASW,UBASWL
      COMMON  VBT(3),VCAP,VI
      COMMON  WFMAAW,WFMASW,WFMPLT,WFMURG,WFTAAW,WFTASW,WFTPLT,WFTURG
      COMMON  WPLNDQ(5),WTFCBO,WVSIZ,WFPPAS(2,5),WFTFL(5)
      COMMON  XAEW,XAEWLQ(5),XASW,XASWLQ(5),XATTCK,XEAAW,XEASWA,XEASWN
      COMMON  XFGHTR,XPLAT,XURGS,XIA(5),XIE(5),XNRAB
      COMMON  ZLAMPF,ZMPCAP,ZMPDLI,ZMPATT(3),ZMPESC,ZMPSTG
C
```

PROGRAM INP

```
      C* COMDECK COMIGO
            COMMON/COMIGO/ IGO
      C*
      C
            DIMENSION IVARQ(512,8),LISTV(512),MPOW(8)
            DIMENSIONIVRB(35),IVRBST(35,50),LISTW(50)
            DIMENSION NVPFMT(4),INFO(8)
            DIMENSION IFMT(2),IF1(2,4),IREC(21),ZREC(21)
            DIMENSION INUMB(4),IOR(2)
            DIMENSION PDIV(1),SDIV(1),ITD(1),IBALD(1)
            EQUIVALENCE (IREC(1),ZREC(1)),(IREC(4),IST),(IREC(5),IMAX),
           1 (IREC(6),INC),(NEPD(1),CEPD)
            DATA(NVPFMT(J),J=1,3)/6,6,7/  ,MBLK,MRED,MUNIT,MB/6HZZZZZZ,1HR,
           $ 4HUNIT,1HB/,NVPFMT(4)/6/
            DATA (IF1(K),K=1,8)/10H(10X,10I11,1H),     10H(10X,10G12,4H.4) ,
           1   1H ,1H , 10H(9X,10A11),1H /
            DATA MO/1H /
            DATA (MPOW(J),J=1,8),N,MPD /128,64,32,16,8,4,2,1,1,0/
            DATA MPNAME/1H /,JDIM/512/,(LISTV(I),I=1,511)/511*0/
            DATA (INUMB(J),J=1,4),IOR(1),IOR(2)/4*12,6HINCREM,6HRPLACE/
            DATA N/ 264/,(IVARQ(I,1),I= 265,512 )/ 248*6HZZZZZZ/
            DATA(IVARQ(  1,K),K=1,8)/6HAAAEDA,  2,  0, 0,     1,   0,20,    31/
            DATA(IVARQ(  2,K),K=1,8)/6HAAAEDE,  2,  0, 0,     3,   0,20,    51/
            DATA(IVARQ(  3,K),K=1,8)/6HAAAEED,  1,  0, 0,     5,   0,20,    61/
            DATA(IVARQ(  4,K),K=1,8)/6HAACA  ,  1,  0, 0,     6,   0,20,    71/
            DATA(IVARQ(  5,K),K=1,8)/6HAAPAJO,  2,  0, 0,     7,   0,20,    91/
            DATA(IVARQ(  6,K),K=1,8)/6HAAPDDA,  2,  0, 0,     9,   0,20,   111/
            DATA(IVARQ(  7,K),K=1,8)/6HAAPDDE,  2,  0, 0,    11,   0,20,   131/
            DATA(IVARQ(  8,K),K=1,8)/6HAAPDED,  1,  0, 0,    13,   0,20,   141/
            DATA(IVARQ(  9,K),K=1,8)/6HAAPKAD,  2,  2, 0,    14,   0,20,   182/
            DATA(IVARQ( 10,K),K=1,8)/6HAAPKDA,  2,  2, 0,    18,   0,20,   222/
            DATA(IVARQ( 11,K),K=1,8)/6HAAPKDE,  2,  1, 0,    22,   0,20,   242/
            DATA(IVARQ( 12,K),K=1,8)/6HAAPKED,  1,  2, 0,    24,   0,20,   262/
            DATA(IVARQ( 13,K),K=1,8)/6HAASRAA,  5,  0, 0,    26,   0,20,   311/
            DATA(IVARQ( 14,K),K=1,8)/6HAASRED,  1,  0, 0,    31,   0,20,   321/
            DATA(IVARQ( 15,K),K=1,8)/6HAASRFA,  5,  0, 0,    32,   0,20,   371/
            DATA(IVARQ( 16,K),K=1,8)/6HAASRFE,  5,  0, 0,    37,   0,20,   421/
            DATA(IVARQ( 17,K),K=1,8)/6HAASRID,  1,  0, 0,    42,   0,20,   431/
            DATA(IVARQ( 18,K),K=1,8)/6HABANM ,  2,  0, 0,    43,   0,21,   451/
            DATA(IVARQ( 19,K),K=1,8)/6HABAVLS,  2,  0, 0,    45,   0,20,   471/
            DATA(IVARQ( 20,K),K=1,8)/6HABCAS ,  1,  0, 0,    47,   0,20,   481/
            DATA(IVARQ( 21,K),K=1,8)/6HABESGS,  2,  0, 0,    48,   0,20,   501/
            DATA(IVARQ( 22,K),K=1,8)/6HABFASS,  2,  0, 0,    50,   0,20,   521/
            DATA(IVARQ( 23,K),K=1,8)/6HABFSM ,  2,  0, 0,    52,   0,20,   541/
            DATA(IVARQ( 24,K),K=1,8)/6HABFVS ,  2,  0, 0,    54,   0,20,   561/
            DATA(IVARQ( 25,K),K=1,8)/6HABPDA ,  2,  0, 0,    56,   0,20,   581/
            DATA(IVARQ( 26,K),K=1,8)/6HABPDS ,  2,  0, 0,    64,   0,20,   661/
            DATA(IVARQ( 27,K),K=1,8)/6HABPKA ,  2,  0, 0,    58,   0,20,   601/
            DATA(IVARQ( 28,K),K=1,8)/6HABPKS ,  2,  2, 0,    66,   0,20,   702/
            DATA(IVARQ( 29,K),K=1,8)/6HABPSA ,  2,  2, 0,    60,   0,20,   642/
            DATA(IVARQ( 30,K),K=1,8)/6HABRSAM,  2,  0, 0,    74,   0,21,   761/
            DATA(IVARQ( 31,K),K=1,8)/6HABTSC ,  2,  0, 0,    70,   0,20,   721/
            DATA(IVARQ( 32,K),K=1,8)/6HABVGSS,  2,  0, 0,    72,   0,20,   741/
            DATA(IVARQ( 33,K),K=1,8)/6HAESCAB,  2,  0, 0,    77,   0,21,   791/
            DATA(IVARQ( 34,K),K=1,8)/6HAEWD  ,  1,  0, 0,    76,   0,20,   771/
            DATA(IVARQ( 35,K),K=1,8)/6HAINTCT,  1,  0, 0,    97,   0,21,   981/
            DATA(IVARQ( 36,K),K=1,8)/6HASWF  ,  1,  0, 0,    79,   0,20,   801/
```

```
DATA(IVARQ( 37,K),K=1,8)/6HATABT ,  2,  3, 0,   80,   0,21,   862/ 15SEP81
DATA(IVARQ( 38,K),K=1,8)/6HATTWGT,  1,  0, 0,   86,   0,20,   871/ 15SEP81
DATA(IVARQ( 39,K),K=1,8)/6HAVAILE,  5,  2, 0,   87,   0,20,   972/ 15SEP81
DATA(IVARQ( 40,K),K=1,8)/6HAVAILT,  5,  2, 3,   98,   0,20,  1283/ 15SEP81
DATA(IVARQ( 41,K),K=1,8)/6HAVALED,  5,  2, 0,  128,   0,20,  1382/ 15SEP81
DATA(IVARQ( 42,K),K=1,8)/6HAWRCBB,  1,  0, 0,  138,   0,20,  1391/ 15SEP81
DATA(IVARQ( 43,K),K=1,8)/6HBACCDW,  6,  0, 0,  139,   0,20,  1451/ 15SEP81
DATA(IVARQ( 44,K),K=1,8)/6HBACPCK,  6,  0, 0,  145,   0,20,  1511/ 15SEP81
DATA(IVARQ( 45,K),K=1,8)/6HBAREAQ,  5,  0, 0,  151,   0,20,  1561/ 15SEP81
DATA(IVARQ( 46,K),K=1,8)/6HBARELQ,  5,  0, 0,  156,   0,20,  1611/ 15SEP81
DATA(IVARQ( 47,K),K=1,8)/6HBARLQ ,  5,  0, 0,  161,   0,20,  1661/ 15SEP81
DATA(IVARQ( 48,K),K=1,8)/6HBARLTH,  5,  0, 0,  171,   0,20,  1761/ 15SEP81
DATA(IVARQ( 49,K),K=1,8)/6HBECDW ,  6,  0, 0,  176,   0,20,  1821/ 15SEP81
DATA(IVARQ( 50,K),K=1,8)/6HBEDW  , 10,  0, 0,  182,   0,20,  1921/ 15SEP81
DATA(IVARQ( 51,K),K=1,8)/6HBMTMIN,  5,  0, 0,  166,   0,20,  1711/ 15SEP81
DATA(IVARQ( 52,K),K=1,8)/6HBSIBAR,  5,  0, 0,  192,   0,21,  1971/ 15SEP81
DATA(IVARQ( 53,K),K=1,8)/6HBSSNDS,  1,  0, 0,  197,   0,21,  1981/ 15SEP81
DATA(IVARQ( 54,K),K=1,8)/6HBUCAP ,  1,  0, 0,  198,   0,20,  1991/ 15SEP81
DATA(IVARQ( 55,K),K=1,8)/6HCACDWO,  1,  0, 0,  199,   0,20,  2001/ 15SEP81
DATA(IVARQ( 56,K),K=1,8)/6HCAPMLQ,  5,  0, 0,  200,   0,20,  2051/ 15SEP81
DATA(IVARQ( 57,K),K=1,8)/6HCAPMQ ,  5,  0, 0,  205,   0,20,  2101/ 15SEP81
DATA(IVARQ( 58,K),K=1,8)/6HCAPMR ,  1,  0, 0,  210,   0,20,  2111/ 15SEP81
DATA(IVARQ( 59,K),K=1,8)/6HCAPSTQ,  5,  0, 0,  211,   0,20,  2161/ 15SEP81
DATA(IVARQ( 60,K),K=1,8)/6HCPAGV ,  1,  0, 0,  216,   0,20,  2171/ 15SEP81
DATA(IVARQ( 61,K),K=1,8)/6HCPBPK ,  6,  0, 0,  217,   0,20,  2231/ 15SEP81
DATA(IVARQ( 62,K),K=1,8)/6HCPBSCK, 10,  0, 0,  223,   0,20,  2331/ 15SEP81
DATA(IVARQ( 63,K),K=1,8)/6HCPRPK , 10,  0, 0,  233,   0,20,  2431/ 15SEP81
DATA(IVARQ( 64,K),K=1,8)/6HCPRSCK,  6,  0, 0,  243,   0,20,  2491/ 15SEP81
DATA(IVARQ( 65,K),K=1,8)/6HCSCDWO,  1,  0, 0,  249,   0,20,  2501/ 15SEP81
DATA(IVARQ( 66,K),K=1,8)/6HDDFAC , 10,  0, 0,  250,   0,20,  2601/ 15SEP81
DATA(IVARQ( 67,K),K=1,8)/6HDDPKC , 10,  0, 0,  260,   0,20,  2701/ 15SEP81
DATA(IVARQ( 68,K),K=1,8)/6HDDPKS , 10,  0, 0,  270,   0,20,  2801/ 15SEP81
DATA(IVARQ( 69,K),K=1,8)/6HDDRKAA, 10,  0, 0,  280,   0,20,  2901/ 15SEP81
DATA(IVARQ( 70,K),K=1,8)/6HDDRKBA, 10,  0, 0,  290,   0,20,  3001/ 15SEP81
DATA(IVARQ( 71,K),K=1,8)/6HDDRSA , 10,  0, 0,  300,   0,20,  3101/ 15SEP81
DATA(IVARQ( 72,K),K=1,8)/6HDDSPA , 10,  0, 0,  310,   0,20,  3201/ 15SEP81
DATA(IVARQ( 73,K),K=1,8)/6HDLIA  ,  1,  0, 0,  320,   0,20,  3211/ 15SEP81
DATA(IVARQ( 74,K),K=1,8)/6HD1T   ,  2,  3, 0,  321,   0,20,  3272/ 15SEP81
DATA(IVARQ( 75,K),K=1,8)/6HD2T   ,  2,  3, 0,  327,   0,20,  3332/ 15SEP81
DATA(IVARQ( 76,K),K=1,8)/6HENACDS, 10,  0, 0,  343,   0,20,  3531/ 15SEP81
DATA(IVARQ( 77,K),K=1,8)/6HENACDT,  4,  0, 0,  339,   0,20,  3431/ 15SEP81
DATA(IVARQ( 78,K),K=1,8)/6HESLR  ,  1,  0, 0,  333,   0,20,  3341/ 15SEP81
DATA(IVARQ( 79,K),K=1,8)/6HESRQ  ,  5,  0, 0,  334,   0,20,  3391/ 15SEP81
DATA(IVARQ( 80,K),K=1,8)/6HFAACA ,  5,  0, 0,  353,   0,20,  3581/ 15SEP81
DATA(IVARQ( 81,K),K=1,8)/6HFACDB ,  5,  2, 0,  368,   0,20,  3782/ 15SEP81
DATA(IVARQ( 82,K),K=1,8)/6HFFACA ,  5,  0, 0,  358,   0,20,  3631/ 15SEP81
DATA(IVARQ( 83,K),K=1,8)/6HFFACE ,  5,  0, 0,  363,   0,20,  3681/ 15SEP81
DATA(IVARQ( 84,K),K=1,8)/6HFHSK  ,  2,  0, 0,  378,   0,20,  3801/ 15SEP81
DATA(IVARQ( 85,K),K=1,8)/6HFM3   ,  6,  0, 0,  380,   0,20,  3861/ 15SEP81
DATA(IVARQ( 86,K),K=1,8)/6HFPPL1 ,  1,  0, 0,  386,   0,20,  3871/ 15SEP81
DATA(IVARQ( 87,K),K=1,8)/6HFPPL2 ,  1,  0, 0,  387,   0,20,  3881/ 15SEP81
DATA(IVARQ( 88,K),K=1,8)/6HFSTAQ ,  5,  0, 0,  388,   0,20,  3931/ 15SEP81
DATA(IVARQ( 89,K),K=1,8)/6HFSTGAQ,  5,  0, 0,  393,   0,20,  3981/ 15SEP81
DATA(IVARQ( 90,K),K=1,8)/6HHRMAAW,  1,  0, 0,  398,   0,20,  3991/ 15SEP81
DATA(IVARQ( 91,K),K=1,8)/6HHRMASW,  1,  0, 0,  399,   0,20,  4001/ 15SEP81
DATA(IVAPQ( 92,K),K=1,8)/6HHRMURG,  1,  0, 0,  400,   0,20,  4011/ 15SEP81
DATA(IVARQ( 93,K),K=1,8)/6HHRTAAW,  1,  0, 0,  401,   0,20,  4021/ 15SEP81
```

```
      DATA(IVARQ( 94,K),K=1,8)/6HHRTASW,  1,  0, 0,  402,   0,20,  4031/ 15SEP81
      DATA(IVARQ( 95,K),K=1,8)/6HHRTURG,  1,  0, 0,  403,   0,20,  4041/ 15SEP81
      DATA(IVARQ( 96,K),K=1,8)/6HIAADA ,  1,  0, 0,  404,   0,10,  4051/ 15SEP81
      DATA(IVARQ( 97,K),K=1,8)/6HIAAED ,  1,  0, 0,  405,   0,10,  4061/ 15SEP81
      DATA(IVARQ( 98,K),K=1,8)/6HIABAEQ,  1,  0, 0,  408,   0,10,  4091/ 15SEP81
      DATA(IVARQ( 99,K),K=1,8)/6HIABAF ,  1,  0, 0,  406,   0,10,  4071/ 15SEP81
      DATA(IVARQ(100,K),K=1,8)/6HIABAW ,  1,  0, 0,  407,   0,10,  4081/ 15SEP81
      DATA(IVARQ(101,K),K=1,8)/6HIATKRT,  5,  0, 0,  409,   0,10,  4141/ 15SEP81
      DATA(IVARQ(102,K),K=1,8)/6HIATRIA,  1,  0, 0,  414,   0,10,  4151/ 15SEP81
      DATA(IVARQ(103,K),K=1,8)/6HICTL  ,  5,  0, 0,  415,   0,10,  4201/ 15SEP81
      DATA(IVARQ(104,K),K=1,8)/6HIDDAC ,  1,  0, 0,  420,   0,10,  4211/ 15SEP81
      DATA(IVARQ(105,K),K=1,8)/6HIDDAS ,  1,  0, 0,  421,   0,10,  4221/ 15SEP81
      DATA(IVARQ(106,K),K=1,8)/6HIKRAS ,  5,  0, 0,  422,   0,10,  4271/ 15SEP81
      DATA(IVARQ(107,K),K=1,8)/6HIPLADA,  1,  0, 0,  427,   0,10,  4281/ 15SEP81
      DATA(IVARQ(108,K),K=1,8)/6HIPLAED,  1,  0, 0,  428,   0,10,  4291/ 15SEP81
      DATA(IVARQ(109,K),K=1,8)/6HIPPAF ,  1,  0, 0,  436,   0,10,  4371/ 15SEP81
      DATA(IVARQ(110,K),K=1,8)/6HIPPAW ,  1,  0, 0,  437,   0,10,  4381/ 15SEP81
      DATA(IVARQ(111,K),K=1,8)/6HIRSUBA,  5,  0, 0,  429,   0,10,  4341/ 15SEP81
      DATA(IVARQ(112,K),K=1,8)/6HISSBR ,  1,  0, 0,  434,   0,10,  4351/ 15SEP81
      DATA(IVARQ(113,K),K=1,8)/6HISSRB ,  1,  0, 0,  435,   0,10,  4361/ 15SEP81
      DATA(IVARQ(114,K),K=1,8)/6HLGTHMP,  6,  0, 0,  438,   0,10,  4441/ 15SEP81
      DATA(IVARQ(115,K),K=1,8)/6HLTEMP ,  6,  0, 0,  444,   0,10,  4501/ 15SEP81
      DATA(IVARQ(116,K),K=1,8)/6HMAXTP ,  1,  0, 0,  450,   0,10,  4511/ 15SEP81
      DATA(IVARQ(117,K),K=1,8)/6HMIMP  ,  1,  0, 0,  451,   0,10,  4521/ 15SEP81
      DATA(IVARQ(118,K),K=1,8)/6HNABSAM,  1,  0, 0,  452,   0,10,  4531/ 15SEP81
      DATA(IVARQ(119,K),K=1,8)/6HNEPD  ,  1,  0, 0,    0,   0,10,    11/ 15SEP81
      DATA(IVARQ(120,K),K=1,8)/6HNKBDPL,  1,  0, 0,  455,   0,10,  4561/ 15SEP81
      DATA(IVARQ(121,K),K=1,8)/6HNKRB  ,  1,  0, 0,  453,   0,10,  4541/ 15SEP81
      DATA(IVARQ(122,K),K=1,8)/6HNKRS  ,  1,  0, 0,  454,   0,10,  4551/ 15SEP81
      DATA(IVARQ(123,K),K=1,8)/6HNLOC  ,  1,  0, 0,  456,   0,10,  4571/ 15SEP81
      DATA(IVARQ(124,K),K=1,8)/6HNPPSAM,  1,  0, 0,  457,   0,10,  4581/ 15SEP81
      DATA(IVARQ(125,K),K=1,8)/6HPAFCNF,  1,  0, 0,  884,   0,20,  8851/ 15SEP81
      DATA(IVARQ(126,K),K=1,8)/6HPARK  ,  1,  0, 0,  458,   0,20,  4591/ 15SEP81
      DATA(IVARQ(127,K),K=1,8)/6HPASS  ,  2,  0, 0,  459,   0,20,  4611/ 15SEP81
      DATA(IVARQ(128,K),K=1,8)/6HPBDRN ,  2,  0, 0,  461,   0,20,  4631/ 15SEP81
      DATA(IVARQ(129,K),K=1,8)/6HPBDRS ,  2,  0, 0,  463,   0,20,  4651/ 15SEP81
      DATA(IVARQ(130,K),K=1,8)/6HPBKRN ,  2,  0, 0,  465,   0,20,  4671/ 15SEP81
      DATA(IVARQ(131,K),K=1,8)/6HPBKRS ,  2,  0, 0,  467,   0,20,  4691/ 15SEP81
      DATA(IVARQ(132,K),K=1,8)/6HPDIN  ,  1,  0, 0,  469,   0,20,  4701/ 15SEP81
      DATA(IVARQ(133,K),K=1,8)/6HPFFCNF,  1,  0, 0,  885,   0,20,  8861/ 15SEP81
      DATA(IVARQ(134,K),K=1,8)/6HPKASW ,  1,  0, 0,  472,   0,20,  4731/ 15SEP81
      DATA(IVARQ(135,K),K=1,8)/6HPKAT1 ,  1,  0, 0,  470,   0,20,  4711/ 15SEP81
      DATA(IVARQ(136,K),K=1,8)/6HPKDF1 ,  1,  0, 0,  471,   0,20,  4721/ 15SEP81
      DATA(IVARQ(137,K),K=1,8)/6HPKIIN ,  1,  0, 0,  473,   0,20,  4741/ 15SEP81
      DATA(IVARQ(138,K),K=1,8)/6HPKIN  ,  1,  0, 0,  474,   0,20,  4751/ 15SEP81
      DATA(IVARQ(139,K),K=1,8)/6HPKPLDT,  4,  0, 0,  475,   0,20,  4791/ 15SEP81
      DATA(IVARQ(140,K),K=1,8)/6HPKPL1 ,  1,  0, 0,  479,   0,20,  4801/ 15SEP81
      DATA(IVARQ(141,K),K=1,8)/6HPKPL2 ,  1,  0, 0,  480,   0,20,  4811/ 15SEP81
      DATA(IVARQ(142,K),K=1,8)/6HPKSST ,  4,  0, 0,  481,   0,20,  4851/ 15SEP81
      DATA(IVARQ(143,K),K=1,8)/6HPLAEDA,  2,  0, 0,  790,   0,20,  7921/ 15SEP81
      DATA(IVARQ(144,K),K=1,8)/6HPLAEDE,  2,  0, 0,  792,   0,20,  7941/ 15SEP81
      DATA(IVARQ(145,K),K=1,8)/6HPLAEED,  1,  0, 0,  794,   0,20,  7951/ 15SEP81
      DATA(IVARQ(146,K),K=1,8)/6HPLBLBD,  2,  5, 0,  795,   0,21,  8052/ 15SEP81
      DATA(IVARQ(147,K),K=1,8)/6HPLCA  ,  5,  0, 0,  805,   0,20,  8101/ 15SEP81
      DATA(IVARQ(148,K),K=1,8)/6HPLFDLL,  5,  5, 2,  810,   0,20,  8603/ 15SEP81
      DATA(IVARQ(149,K),K=1,8)/6HPLPAJD,  3,  0, 0,  860,   0,20,  8631/ 15SEP81
      DATA(IVARQ(150,K),K=1,8)/6HPLPDDA,  2,  0, 0,  863,   0,20,  8651/ 15SEP81
```

```
DATA(IVARQ(151,K),K=1,8)/6HPLPDDE,  2,  0, 0,  865,   0,20,  8671/ 15SEP81
DATA(IVARQ(152,K),K=1,8)/6HPLPDED,  1,  0, 0,  867,   0,20,  8681/ 15SEP81
DATA(IVARQ(153,K),K=1,8)/6HPLPKAD,  3,  2, 0,  868,   0,20,  8742/ 15SEP81
DATA(IVARQ(154,K),K=1,8)/6HPLPKDA,  2,  3, 0,  874,   0,20,  8802/ 15SEP81
DATA(IVARQ(155,K),K=1,8)/6HPLPKDE,  2,  0, 0,  880,   0,20,  8821/ 15SEP81
DATA(IVARQ(156,K),K=1,8)/6HPLPKED,  2,  0, 0,  882,   0,20,  8841/ 15SEP81
DATA(IVARQ(157,K),K=1,8)/6HPPAEGS,  2,  0, 0,  937,   0,20,  9391/ 15SEP81
DATA(IVARQ(158,K),K=1,8)/6HPPANMS,  2,  0, 0,  920,   0,21,  9221/ 15SEP81
DATA(IVARQ(159,K),K=1,8)/6HPPAVLS,  2,  5, 0,  910,   0,20,  9202/ 15SEP81
DATA(IVARQ(160,K),K=1,8)/6HPPAVSS,  2,  0, 0,  906,   0,20,  9081/ 15SEP81
DATA(IVARQ(161,K),K=1,8)/6HPPCAL ,  5,  0, 0,  928,   0,20,  9331/ 15SEP81
DATA(IVARQ(162,K),K=1,8)/6HPPFASM,  2,  0, 0,  939,   0,20,  9411/ 15SEP81
DATA(IVARQ(163,K),K=1,8)/6HPPFASS,  2,  0, 0,  935,   0,20,  9371/ 15SEP81
DATA(IVARQ(164,K),K=1,8)/6HPPFSVS,  2,  0, 0,  924,   0,20,  9261/ 15SEP81
DATA(IVARQ(165,K),K=1,8)/6HPPPDAS,  2,  0, 0,  933,   0,20,  9351/ 15SEP81
DATA(IVARQ(166,K),K=1,8)/6HPPPDSA,  2,  0, 0,  922,   0,20,  9241/ 15SEP81
DATA(IVARQ(167,K),K=1,8)/6HPPPKAS,  2,  0, 0,  908,   0,20,  9101/ 15SEP81
DATA(IVARQ(168,K),K=1,8)/6HPPPKSA,  2,  2, 0,  900,   0,20,  9042/ 15SEP81
DATA(IVARQ(169,K),K=1,8)/6HPPPSAS,  2,  2, 0,  896,   0,20,  9002/ 15SEP81
DATA(IVARQ(170,K),K=1,8)/6HPPRSAM,  2,  0, 0,  904,   0,21,  9061/ 15SEP81
DATA(IVARQ(171,K),K=1,8)/6HPPSDRR,  2,  5, 0,  886,   0,20,  8962/ 15SEP81
DATA(IVARQ(172,K),K=1,8)/6HPPTSCS,  2,  0, 0,  926,   0,20,  9281/ 15SEP81
DATA(IVARQ(173,K),K=1,8)/6HPRSM  , 10,  5, 6,  485,   0,20,  7853/ 15SEP81
DATA(IVARQ(174,K),K=1,8)/6HPRWLNQ,  5,  0, 0,  785,   0,20,  7901/ 15SEP81
DATA(IVARQ(175,K),K=1,8)/6HRACCDW, 10,  0, 0,  941,   0,20,  9511/ 15SEP81
DATA(IVARQ(176,K),K=1,8)/6HRACPCK, 10,  0, 0,  951,   0,20,  9611/ 15SEP81
DATA(IVARQ(177,K),K=1,8)/6HRARBAB,  3,  0, 0,  977,   0,20,  9801/ 15SEP81
DATA(IVARQ(178,K),K=1,8)/6HRECDW , 10,  0, 0,  961,   0,20,  9711/ 15SEP81
DATA(IVARQ(179,K),K=1,8)/6HREDW  ,  6,  0, 0,  971,   0,20,  9771/ 15SEP81
DATA(IVARQ(180,K),K=1,8)/6HRSIBAR,  5,  0, 0, 1030,   0,21, 10351/ 15SEP81
DATA(IVARQ(181,K),K=1,8)/6HRS    , 10,  5, 0,  980,   0,21, 10302/ 15SEP81
DATA(IVARQ(182,K),K=1,8)/6HSBFBCF,  1,  0, 0, 1035,   0,20, 10361/ 15SEP81
DATA(IVARQ(183,K),K=1,8)/6HSBFBCS,  1,  0, 0, 1036,   0,20, 10371/ 15SEP81
DATA(IVARQ(184,K),K=1,8)/6HSBFRFA,  5,  0, 0, 1037,   0,20, 10421/ 15SEP81
DATA(IVARQ(185,K),K=1,8)/6HSBFRFC,  1,  0, 0, 1042,   0,20, 10431/ 15SEP81
DATA(IVARQ(186,K),K=1,8)/6HSBFRSA,  5,  0, 0, 1043,   0,20, 10481/ 15SEP81
DATA(IVARQ(187,K),K=1,8)/6HSBFRSC,  1,  0, 0, 1048,   0,20, 10491/ 15SEP81
DATA(IVARQ(188,K),K=1,8)/6HSBPBDF,  1,  0, 0, 1049,   0,20, 10501/ 15SEP81
DATA(IVARQ(189,K),K=1,8)/6HSBPBDS,  1,  0, 0, 1050,   0,20, 10511/ 15SEP81
DATA(IVARQ(190,K),K=1,8)/6HSBPBKF,  1,  0, 0, 1051,   0,20, 10521/ 15SEP81
DATA(IVARQ(191,K),K=1,8)/6HSBPBKS,  1,  0, 0, 1052,   0,20, 10531/ 15SEP81
DATA(IVARQ(192,K),K=1,8)/6HSBPFDB,  1,  0, 0, 1053,   0,20, 10541/ 15SEP81
DATA(IVARQ(193,K),K=1,8)/6HSBPFKB,  1,  0, 0, 1054,   0,20, 10551/ 15SEP81
DATA(IVARQ(194,K),K=1,8)/6HSBPSDB,  1,  0, 0, 1055,   0,20, 10561/ 15SEP81
DATA(IVARQ(195,K),K=1,8)/6HSBPSKB,  1,  0, 0, 1056,   0,20, 10571/ 15SEP81
DATA(IVARQ(196,K),K=1,8)/6HSHEL  ,  1,  0, 0, 1068,   0,21, 10691/ 15SEP81
DATA(IVARQ(197,K),K=1,8)/6HSMALLR,  1,  0, 0, 1057,   0,20, 10581/ 15SEP81
DATA(IVARQ(198,K),K=1,8)/6HSSBACR,  8,  0, 0, 1069,   0,20, 10771/ 15SEP81
DATA(IVARQ(199,K),K=1,8)/6HSSCFA ,  1,  0, 0, 1077,   0,20, 10781/ 15SEP81
DATA(IVARQ(200,K),K=1,8)/6HSSDAAW,  1,  0, 0, 1058,   0,20, 10591/ 15SEP81
DATA(IVARQ(201,K),K=1,8)/6HSSDASW,  1,  0, 0, 1059,   0,20, 10601/ 15SEP81
DATA(IVARQ(202,K),K=1,8)/6HSSDURG,  1,  0, 0, 1060,   0,20, 10611/ 15SEP81
DATA(IVARQ(203,K),K=1,8)/6HSSFBAK,  2,  8, 0, 1122,   0,20, 11382/ 15SEP81
DATA(IVARQ(204,K),K=1,8)/6HSSFRSV,  8,  5, 0, 1078,   0,20, 11182/ 15SEP81
DATA(IVARQ(205,K),K=1,8)/6HSSPBDR,  1,  0, 0, 1118,   0,20, 11191/ 15SEP81
DATA(IVARQ(206,K),K=1,8)/6HSSPBKR,  1,  0, 0, 1119,   0,20, 11201/ 15SEP81
DATA(IVARQ(207,K),K=1,8)/6HSSPRDB,  1,  0, 0, 1120,   0,20, 11211/ 15SEP81
```

```
DATA(IVARQ(208,K),K=1,8)/6HSSPRKB,  1,  0, 0, 1121,   0,20, 11221/ 15SEP81   326
DATA(IVARQ(209,K),K=1,8)/6HSSPRKC,  1,  0, 0, 1138,   0,20, 11391/ 15SEP81   327
DATA(IVARQ(210,K),K=1,8)/6HSTARQ ,  5,  0, 0, 1061,   0,20, 10661/ 15SEP81   328
DATA(IVARQ(211,K),K=1,8)/6HSTSALV,  1,  0, 0, 1066,   0,20, 10671/ 15SEP81   329
DATA(IVARQ(212,K),K=1,8)/6HSUBSDR,  1,  0, 0, 1067,   0,20, 10681/ 15SEP81   330
DATA(IVARQ(213,K),K=1,8)/6HTAB10T, 20,  4, 0, 1139,   0,20, 12192/ 15SEP81   331
DATA(IVARQ(214,K),K=1,8)/6HTAB12 , 20,  0, 0, 1219,   0,20, 12391/ 15SEP81   332
DATA(IVARQ(215,K),K=1,8)/6HTAB13T, 20,  4, 0, 1239,   0,20, 13192/ 15SEP81   333
DATA(IVARQ(216,K),K=1,8)/6HTCAP  ,  1,  0, 0, 1319,   0,20, 13201/ 15SEP81   334
DATA(IVARQ(217,K),K=1,8)/6HTHSCAQ,  5,  0, 0, 1320,   0,20, 13251/ 15SEP81   335
DATA(IVARQ(218,K),K=1,8)/6HTHSCTQ,  5,  0, 0, 1325,   0,20, 13301/ 15SEP81   336
DATA(IVARQ(219,K),K=1,8)/6HTPAS  ,  1,  0, 0, 1330,   0,20, 13311/ 15SEP81   337
DATA(IVARQ(220,K),K=1,8)/6HTPS   ,  1,  0, 0, 1331,   0,20, 13321/ 15SEP81   338
DATA(IVARQ(221,K),K=1,8)/6HT1    ,  1,  0, 0, 1332,   0,20, 13331/ 15SEP81   339
DATA(IVARQ(222,K),K=1,8)/6HT2    ,  1,  0, 0, 1333,   0,20, 13341/ 15SEP81   340
DATA(IVARQ(223,K),K=1,8)/6HT3    ,  1,  0, 0, 1334,   0,20, 13351/ 15SEP81   341
DATA(IVARQ(224,K),K=1,8)/6HT4    ,  1,  0, 0, 1335,   0,20, 13361/ 15SEP81   342
DATA(IVARQ(225,K),K=1,8)/6HUBAEWL,  1,  0, 0, 1337,   0,20, 13381/ 15SEP81   343
DATA(IVARQ(226,K),K=1,8)/6HUBAEW ,  1,  0, 0, 1336,   0,20, 13371/ 15SEP81   344
DATA(IVARQ(227,K),K=1,8)/6HUBASWL,  1,  0, 0, 1339,   0,20, 13401/ 15SEP81   345
DATA(IVARQ(228,K),K=1,8)/6HUBASW ,  1,  0, 0, 1338,   0,20, 13391/ 15SEP81   346
DATA(IVARQ(229,K),K=1,8)/6HVBT   ,  3,  0, 0, 1340,   0,20, 13431/ 15SEP81   347
DATA(IVARQ(230,K),K=1,8)/6HVCAP  ,  1,  0, 0, 1343,   0,20, 13441/ 15SEP81   348
DATA(IVARQ(231,K),K=1,8)/6HVI    ,  1,  0, 0, 1344,   0,20, 13451/ 15SEP81   349
DATA(IVARQ(232,K),K=1,8)/6HWFMAAW,  1,  0, 0, 1345,   0,20, 13461/ 15SEP81   350
DATA(IVARQ(233,K),K=1,8)/6HWFMASW,  1,  0, 0, 1346,   0,20, 13471/ 15SEP81   351
DATA(IVARQ(234,K),K=1,8)/6HWFMPLT,  1,  0, 0, 1347,   0,20, 13481/ 15SEP81   352
DATA(IVARQ(235,K),K=1,8)/6HWFMURG,  1,  0, 0, 1348,   0,20, 13491/ 15SEP81   353
DATA(IVARQ(236,K),K=1,8)/6HWFPPAS,  2,  5, 0, 1360,   0,20, 13702/ 15SEP81   354
DATA(IVARQ(237,K),K=1,8)/6HWFTAAW,  1,  0, 0, 1349,   0,20, 13501/ 15SEP81   355
DATA(IVARQ(238,K),K=1,8)/6HWFTASW,  1,  0, 0, 1350,   0,20, 13511/ 15SEP81   356
DATA(IVARQ(239,K),K=1,8)/6HWFTFL ,  5,  0, 0, 1370,   0,20, 13751/ 15SEP81   357
DATA(IVARQ(240,K),K=1,8)/6HWFTPLT,  1,  0, 0, 1351,   0,20, 13521/ 15SEP81   358
DATA(IVARQ(241,K),K=1,8)/6HWFTURG,  1,  0, 0, 1352,   0,20, 13531/ 15SEP81   359
DATA(IVARQ(242,K),K=1,8)/6HWRLNDQ,  5,  0, 0, 1353,   0,20, 13581/ 15SEP81   360
DATA(IVARQ(243,K),K=1,8)/6HWTFCBO,  1,  0, 0, 1358,   0,20, 13591/ 15SEP81   361
DATA(IVARQ(244,K),K=1,8)/6HWVSIZ ,  1,  0, 0, 1359,   0,20, 13601/ 15SEP81   362
DATA(IVARQ(245,K),K=1,8)/6HXAEWLQ,  5,  0, 0, 1376,   0,21, 13811/ 15SEP81   363
DATA(IVARQ(246,K),K=1,8)/6HXAEW  ,  1,  0, 0, 1375,   0,21, 13761/ 15SEP81   364
DATA(IVARQ(247,K),K=1,8)/6HXASWLQ,  5,  0, 0, 1382,   0,21, 13871/ 15SEP81   365
DATA(IVARQ(248,K),K=1,8)/6HXASW  ,  1,  0, 0, 1381,   0,21, 13821/ 15SEP81   366
DATA(IVARQ(249,K),K=1,8)/6HXATTCK,  1,  0, 0, 1387,   0,21, 13881/ 15SEP81   367
DATA(IVARQ(250,K),K=1,8)/6HXEAAW ,  1,  0, 0, 1388,   0,21, 13891/ 15SEP81   368
DATA(IVARQ(251,K),K=1,8)/6HXEASWA,  1,  0, 0, 1389,   0,21, 13901/ 15SEP81   369
DATA(IVARQ(252,K),K=1,8)/6HXEASWN,  1,  0, 0, 1390,   0,21, 13911/ 15SEP81   370
DATA(IVARQ(253,K),K=1,8)/6HXFGHTR,  1,  0, 0, 1391,   0,21, 13921/ 15SEP81   371
DATA(IVARQ(254,K),K=1,8)/6HXIA   ,  5,  0, 0, 1394,   0,20, 13991/ 15SEP81   372
DATA(IVARQ(255,K),K=1,8)/6HXIE   ,  5,  0, 0, 1399,   0,20, 14041/ 15SEP81   373
DATA(IVARQ(256,K),K=1,8)/6HXNRAB ,  1,  0, 0, 1404,   0,20, 14051/ 15SEP81   374
DATA(IVARQ(257,K),K=1,8)/6HXPLAT ,  1,  0, 0, 1392,   0,21, 13931/ 15SEP81   375
DATA(IVARQ(258,K),K=1,8)/6HXURGS ,  1,  0, 0, 1393,   0,21, 13941/ 15SEP81   376
DATA(IVARQ(259,K),K=1,8)/6HZLAMPF,  1,  0, 0, 1405,   0,20, 14061/ 15SEP81   377
DATA(IVARC(260,K),K=1,8)/6HZMPATT,  3,  0, 0, 1408,   0,20, 14111/ 15SEP81   378
DATA(IVARQ(261,K),K=1,8)/6HZMPCAP,  1,  0, 0, 1406,   0,20, 14071/ 15SEP81   379
DATA(IVARQ(262,K),K=1,8)/6HZMPDLI,  1,  0, 0, 1407,   0,20, 14081/ 15SEP81   380
DATA(IVARQ(263,K),K=1,8)/6HZMPESC,  1,  0, 0, 1411,   0,20, 14121/ 15SEP81   381
DATA(IVARQ(264,K),K=1,8)/6HZMPSTG,  1,  0, 0, 1412,   0,20, 14131/ 15SEP81   382
```

```
C
C      LIST OF ITEMS REQUIRING SPECIAL ATTENTION IN CONVERSION
C       1)   ENCODE / DECODE STATEMENTS
C     2) WORD LENGTH W/R IVARQ,INFO,IF1,IFMT,MBLK,NAME
C     3)    DAY CONVERSION ROUTINE
C     4)      G FORMAT SPECIFICATION
C
C
C         INP READS IN TWO (2) SETS OF DATA CARDS.  COMMON IS DEFINED
C         BY DATA STATEMENTS. THE FIRST SET OF CARDS IS THE DEFINITIONS
C         OF THE VARIABLES IN COMMON. THE SECOND SET GIVES THE VALUES THE
C         VARIABLES WILL HAVE. BOTH SETS MUST HAVE A CARD WITH ZZZZZZ IN
C         COLUMNS 1-6 AS A DELIMITER.
C
      DO22I=1,N
      L=IVARQ(I,5)+1
      L1=IVARQ(I,8)/10
      IF(I.EQ.N)L1=MPD
      K=0
      IF(IVARQ(I,7)/10.EQ.4)K=MO
      DO22J=L,L1
   22 NEPD(J)=K
      MOT=6
      IPT=7
      MCF=8
C
C         READ IN THE VARIABLE DESCRIPTIONS, UP TO 5 CARDS A VARIABLE
C         WITH THE FOLLOWING FORMAT:
C              COL 1-6 = NAME OF VARIABLE (NAME)
C              COL 7 = SEQUENCE NUMBER OF CARD (1-5) (I)
C              COL 8-77 = DEFINITION OF VARIABLE TO BE PRINTED OUT (INFO)
C         THESE DESCRIPTIONS ARE ASSUMED TO BE IN ALPHABETICAL ORDER.
C
      ISEQ=0
      JSEQ=1
 1301 READ   1310,NAME,I,(INFO(J),J=1,7)
 1310 FORMAT(A6,I1,7A10)
C
C         MPNAME IS INITIALLY BLANK, BUT WILL LATER BE CHANGED,MO IS THE
C         CONSTANT BLANK. SO FOR THE FIRST TIME THROUGH NAME WILL NOT
C         EQUAL A BLANK AND MPNAME WILL EQUAL A BLANK SO CONTROL GOES TO
C         1303 WHERE IVRB WILL BE SET TO ALL BLANKS, MPNAME IS SET TO
C         NAME, AND SINCE NAME WILL NOT BE EQUAL TO MBLK (ZZZZZZ) CONTROL
C         WILL FALL THROUGH TO THE SEARCH ROUTINE.
C         WHEN NAME HAS BEEN FOUND INFO WILL BE STORED IN THE
C         CORRECT PLACE IN IVRB AND ANOTHER CARD WILL BE READ IN. IF THE
C         DESCRIPTION OF THE VARIABLE IS MORE THAN ONE CARD LONG NAME
C         WILL NOW EQUAL MPNAME AND CONTROL WILL GO DIRECTLY TO 1302 TO
C         STORE INFO. WHEN THE WHOLE DESCRIPTION HAS BEEN READ IN NAME
C         WILL NOT EQUAL MPNAME, AND MPNAME WILL NOT BE BLANK SO THE
C         SEQUENCE COUNTER IS INCREMENTED.  THIS SEQUENCE NUMBER IS THEN
C         STORED IN LISTV IN PARALLEL TO IVARQ (NOTE: IF NAME WAS THE
C         TENTH DESCRIPTION READ IN ISEQ=10, AND IF NAME WAS FOUND IN
C         IVARQ(25,1), THEN LISTV(25)=10). IVRB IS WRITTEN ONTO TAPE 16.
C         IVRB IS BLANKED OUT,MPNAME IS UPDATED TO THE LAST NAME READ IN
C         AND THE CYCLE IS REPEATED UNTIL THE DELIMITER IS HIT AND CON-
C         TROL GOES TO 1309 WHERE TAPE 16 IS REWOUND.
```

```
C
      IF(NAME.EQ.MPNAME)GOTO1302
      IF(MPNAME.EQ.MO)GOTO1303
      ISEQ=ISEQ+1
      LISTV(JV)=ISEQ
      WRITE(16) IVRB
 1303 DO1304K=1,35
 1304 IVRB(K)=MO
      MPNAME=NAME
      IF(NAME.EQ.MBLK)GOTO1309
C
C        ERROR CHECK, IF NAME IS NOT IN COMMON PRINT AND IGNORE.
C
      JD1=JDIM-1
      IF(JSEQ.GT.JD1) GO TO 1305
      DO 1306 JS=JSEQ,JD1
      IF(NAME.NE.IVARQ(JS,1)) GO TO 1306
      JSEQ=JS+1
      JV=JS
      GO TO 1302
 1306 CONTINUE
C
C        THERE IS AN INVALID DESCRIPTION PRINT OUT MESSAGE
C
 1305 WRITE(MOT,1320) NAME,I,(INFO(J),J=1,7)
 1320 FORMAT(38H UNRECOGNIZED AND IGNORED DECRIPTION ,A6,I1,7A10)
      MPNAME=MO
      GOTO1301
C
C        STORE THE DESCRIPTION (INFO(KK),KK=1,7) INTO THE PROPER 35
C        WORDS OF IVRB KEYING ON I, THE SEQUENCE NUMBER OF THE CARD.
C
 1302 CONTINUE
      I=MINO(5,MAXO(I,1))*7
      J=I-6
      KK=0
      DO1308K=J,I
      KK=KK+1
 1308 IVRB(K)=INFO(KK)
      GOTO1301
 1309 REWIND 16
  101 READ110,NAME,NC,NBQ,I,J,K,INFO
  110 FORMAT(A6,I2,A1,3I3,2X,6A10,T19,2R1)
      IF(NAME.EQ.MBLK)GO TO 999
C     ROUTINE FOR 2-BYTE REPRESENTATION OF DAY NUMBER...DAY 0--99IS
C     REPRESENTED AS 00--99, OR B0--99, ORBB--99.
C     DAY 100 IS A, 110 IS B, ETCEG, DAY 137 IS D7
C      CODE IS FOR CDC SOFTWARE AND MUST BE CONVERTED.
C
C        CONVERT A BLANK TO A ZERO.
C        CONVERT FROM A CHARACTER TO A NUMBER BY SUBTRACTING 27(33
C        OCTAL).  THIS ASSUMES CHARACTER VALUE OF ALPHABETICS STARTS AT
C        01 OCTAL AND NUMBERS AT 33 OCTAL.
C
      IF(INFO(8).EQ.1R ) INFO(8)=1R0
      IDAY=INFO(8)-33B
      IF(INFO(7).EQ.1R ) INFO(7)=1R0
```

```
      IF(INFO(7).LT.1R0.OR.INFO(7).GT.1R9) GO TO 301
      IDAY=IDAY+10*(INFO(7)-33B)
      GO TO 302
  301 IDAY=IDAY+(10*INFO(7))+90
  302 CONTINUE
C      END OF DAY CONVERSION ROUTINE
C
C         USING THE BINARY SEARCH TECHNIQUE IT TAKES A MAXIMUM OF 8
C         COMPARISONS TO FIND NAME IN IVARQ. (NOTE: A LINEAR SEARCH WOULD
C         TAKE AN AVERAGE OF 129 COMPARISONS.
C
      JV=JDIM/2.
      DO 109 JJ=1,8
      IF(IVARQ(JV,1).GT.NAME)GOTO103
      IF(IVARQ(JV,1).EQ.NAME)GOTO121
  102 JV=JV+MPOW(JJ)
      GO TO 109
  103 JV=JV-MPOW(JJ)
  109 CONTINUE
      IF(IVARQ(JV,1).EQ.NAME)GO TO 121
      WRITE(MOT,120) NAME,NC,NBQ,I,J,K,INFO(7),INFO(8),(INFO(L),L=1,6)
  120 FORMAT(40H INPUT NAME NOT IN LIST---CARD IGNORED   ,A6,I2,A1,3I3,
     1       2R1,6A10)
      GO TO 101
  121 NFN=IVARQ(JV,7)/10
      NPC=NVPFMT(NFN)
      KDISP=IVARQ(JV,5)
      MAXI=IVARQ(JV,2)
      MAXK=IVARQ(JV,4)
      MAXJ=IVARQ(JV,3)
      ICL=IVARQ(JV,8)-10*(IVARQ(JV,8)/10)
      IF(ICL.EQ.1.AND.I.GT.0)GOTO2171
      IF(ICL.EQ.1)GOTO134
      IF(ICL.EQ.3)GOTO2132
      IF(I.LE.0.AND.J.LE.0.OR.I.GT.MAXI.OR.J.GT.MAXJ)GOTO2135
      GOTO132
 2132 IF(I.GT.MAXI.OR.J.GT.MAXJ.OR.K.GT.MAXK)GO TO 2135
      IF(I.LE.0.AND.J*K.LE.0.OR.(J.LE.0.AND.K.LE.0))GOTO2135
      GOTO133
 2135 WRITE(MOT,2136) ICL,NAME,NC,NBQ,I,J,K,INFO
 2136 FORMAT(19H IMPROPER CODING OF ,I2,44H DIMENSIONAL VARIABLE -- CARD
     1 BELOW IGNORED ,/,1X,A6,I2,A1,3I3,2X,6A10,T20,2R1)
      GOTO101
  132 IF(I.NE.0)ICL=3
      IF(ICL.EQ.3.AND.J.GT.0)GOTO2172
      GO TO 134
  133 IF(I.EQ.0)ICL=4
      IF(J.EQ.0)ICL=5
      IF(K.EQ.0)ICL=6
      IF(ICL.NE.3)GOTO134
C     SINGLE VALUE DIRECT INPUT WHEN ALL INDEX VALUES ARE SPECIFIED.
C
C       3-DIMENSION CASE
      IST=MAXI*(MAXJ*(K-1)+J-1)+I
 2174 IMAX=IST
      INC=1
      ICL=7
```

```
      GOTO149
C       1-DIMENSION CASE
 2171 IST=I
      GOTO2174
C      2-DIMENSION CASE
 2172 IST=(J-1)*MAXI+I
      GOTO2174
  134 IF(IVARQ(JV,6).EQ.0.OR. NBQ.EQ.MO)GOTO137
      GO TO (201,202,203,204,204,204,204) ,ICL
  201 MAXI=MAXI/2
      GOTO207
  202 MAXJ=MAXJ/2
      IF(J.GT.MAXJ)J=J-MAXJ
      GOTO207
  203 MAXJ=MAXJ/2
      GOTO207
  204 MAXK=MAXK/2
      IF(K.GT.MAXK)K=K-MAXK
  207 IF(NBQ.NE.MRED)GOTO137
      KDISP=KDISP+IVARQ(JV,6)
  137 INC=1
      KQ=NC*NPC+1
      GO TO (141,142,143,144,145,146), ICL
C       READ V(I),I=1,,,
  141 IST=KQ
      IMAX=MIN0(MAXI,IST+NPC-1)
      GO TO 149
C       READ(V(I,J),I=1,,,,,,)
  142 IST=(J-1)*MAXI+KQ
      IMAX=MIN0(J*MAXI,IST+NPC-1)
      GO TO 149
C      READ(V(I,J),J=1,,,,,,,)
  143 IST=I+NPC*NC*MAXI
      INC=MAXI
C      IMAX=IST+INC*MIN0(NPC,MAXJ)-INC
      IMAX=IST+INC*MIN0(NPC,MAXJ-NC*6)-INC
      GO TO 149
C      READ(V(I,J,K,I=1,,,,,,,,,)
  144 IST=MAXI*((K-1)*MAXJ+(J-1))+KQ
      IMAX=IST+MIN0(NPC-1,MAXI-KQ)
      GO TO 149
C       READ(V(I,J,K),J=1,,,,,,,)
  145 INC=MAXI
      IST=I+INC*((K-1)*MAXJ+KQ-1)
      IMAX=IST+INC*(MIN0(NPC,MAXJ-KQ+1)-1)
      GO TO 149
C       READ(V(I,J,K),K=1,,,,,,,)
  146 INC=MAXI*MAXJ
      IST=KQ*INC+MAXI*(J-1)+I-INC
      IMAX=IST+INC*(MIN0(NPC,MAXK-KQ+1)-1)
  149 IST=IST+KDISP
      IMAX=IMAX+KDISP
      IF(IST.LE.IVARQ(JV,5).OR.IMAX.GT.IVARQ(JV,8)/10) GO TO 2135
      IF(IDAY.LE.0)GO TO (151,161,163,163),NFN
C      OUTPUT RECORD FOR LATER PROCESSING
      IF(IDAY.EQ.MPDAY)GO TO 171
      MPDAY=IDAY
```

```
      IREC(1)=9999
      IREC(2)=MPDAY
      WRITE(15)IREC
      IREC(1)=MPDAY
  171 IREC(2)=NFN
      IREC(3)=IVARQ(JV,7)-10*NFN
      KK=K
      LL=IREC(3)
      IF (LL.GT.0) LL=IOR(LL)
      IF (LL.LE.0) LL=6H
      IM=INUMB(NFN)
      IF(ICL.GT.6)IM=7
      IFMT(1)=IF1(1,NFN)
      IFMT(2)=IF1(2,NFN)
      GO TO (172,173,177,177),NFN
  172 DECODE(60,1510,INFO)(IREC(K),K=7,IM)
      GO TO 174
  173 DECODE(60,1610,INFO)(IREC(K),K=7,IM)
  174 WRITE(15)IREC
      WRITE(MOT,2120) IDAY,LL,NAME,NC,NBQ,I,J,KK
      WRITE(MOT,IFMT) (IREC(K),K=7,IM)
 2120 FORMAT(9HOTIME-T= ,I3,A6,10H VARIABLE ,A6,I2,1H ,A1,3I3,20H---VALU
     1ES ARE BELOW )
      GO TO 101
  151 DECODE(60,1510,INFO)(NEPD(I   ),I=IST,IMAX,INC)
      GO TO 101
  161 DECODE(60,1610,INFO)(NEPD(I   ),I=IST,IMAX,INC)
      GO TO 101
  163 DECODE(60,1720,INFO)(NEPD(I),I=IST,IMAX,INC)
      GOTO101
  177 DECODE(60,1720,INFO)(IREC(K),K=7,IM)
      GOTO174
 1720 FORMAT(6A10)
 1510 FORMAT(6I10)
 1610 FORMAT(6F10.0)
  999 DO990 JV=1,N
      WRITE(MOT,980) (IVARQ(JV,K),K=1,4)
  980 FORMAT(15HOVARIABLE ---- ,A6,1H(,I3,2(1H,,I3),1H))
      IF(LISTV(JV).LE.0)GOTO1329
      READ(16)IVRB
      DO1328K=1,29,7
      LL=K+6
      DO1327L=K,LL
      IF(IVRB(L).NE.MO)GOTO1325
 1327 CONTINUE
      GOTO1328
 1325 WRITE(MOT,1330) (IVRB(L),L=K,LL)
 1330 FORMAT(35X,7A10)
 1328 CONTINUE
 1329 CONTINUE
      NFN=IVARQ(JV,7)/10
      KDISP=IVARQ(JV,5)
      ICL=IVARQ(JV,8)-10*(IVARQ(JV,8)/10)
      MAXI=IVARQ(JV,2)
      MAXJ=IVARQ(JV,3)
      MAXK=IVARQ(JV,4)
      IFMT(1)=IF1(1,NFN)
```

```
      IFMT(2)=IF1(2,NFN)
      GOTO(971,972,973),ICL
  971 IST=KDISP+1
      IMAX=KDISP+MAXI
      WRITE(MOT,IFMT) (NEPD(I),I=IST,IMAX)
      GOTO990
  972 INC=MAXI
      ASSIGN 990 TO MGO
 9732 DO9721 K=1,MAXI
      IST=KDISP+K
      JMAX=KDISP+MAXI*MAXJ-MAXI+K
 9721 WRITE(MOT,IFMT) (NEPD(I),I=IST,JMAX,INC)
      GO TO MGO, (990,9731)
  973 ASSIGN 9731 TO MGO
      INC=MAXI
      KK=1
      WRITE(MOT,9730) KK
 9730 FORMAT(4H K = ,I2)
      GO TO 9732
 9731 KK=KK+1
      IF(KK.GT.MAXK)GO TO 990
      KDISP=KDISP+MAXI*MAXJ
      WRITE(MOT,9730) KK
      GO TO 9732
  990 CONTINUE
      END FILE 15
      REWIND 15
      IGO=0
      IF(IDAY.GT.0) READ (15) IGO,IGO
      END
```

DISTRIBUTION LIST, IDA PAPER P-1615, VOLUME III

"Net Assessment Methodologies and Critical Data Elements for Strategic and Theater Force Comparisons for Total Force Capability Assessment (TFCA) (U): Volume III -- A Preliminary Documentation of a Naval Model (Second Interim Report)"

ADDRESSEE		NUMBER OF COPIES
DEPARTMENT OF DEFENSE		
Joint Chiefs of Staff Distribution Branch GAP Division DAS The Pentagon Washington, D.C. 20301		
ATTENTION:	JOINT SECRETARIAT	200*
ATTENTION:	Office of Director, Studies Analysis and Gaming Agency	
	RADM K. S. Masterson, USN (Room 1D936)	1
	Dr. William G. Lese (Room 1D940C)	1
	CAPT James P. O'Neill, USN (Room 1D929A)	12
ATTENTION:	Office of Director, Command, Control and Communications Systems	
	COL P. P. Winkel (Room 1D825)	1
ATTENTION:	Office of the Director, J-3 (Operations)	
	COL J. D. James, USA (Room 2B877B)	2
ATTENTION:	Office of the Director, J-4 (Logistics)	
	COL John S. Foard, USN (Room 2E827)	2
ATTENTION:	Office of the Director, J-5 (Plans & Policy)	
	LTC A. W. Kremer, Jr. USA (Room 1C968B)	2
Defense Intelligence Agency Room 2D233, Pomponio Plaza Washington, D.C. 20301		
ATTENTION:	Office of the Director (Mr. H. J. Tessandori) (Pomponio Plaza 1023)	2

*200 copies and manuscript in accordance with SM-521-82.

ADDRESSEE	NUMBER OF COPIES
DOD-IDA Management Office 1801 North Beauregard Street Alexandria, Virginia 22311	1

<u>DEPARTMENT OF THE ARMY</u>

Office of the Deputy Chief of Staff for
Operations and Plans
U.S. Army (DAMO)
The Pentagon (Room 3C-542)
Washington, D.C. 20310

ATTENTION: MAJ D. R. Nelson, USA (Room 3E543) — 2

<u>DEPARTMENT OF THE NAVY</u>

Chief of Naval Operations
Department of the Navy
Room 4D-541, The Pentagon
Washington, D.C. 20350

ATTENTION: Office of the Deputy Chief of Naval Operations
(Plans, Policy and Operations) (NOP)
CDR Tim Quigley, USN (Room 4-E531) — 2

Headquarters, U.S. Marine Corps
Code HQSM-3, Room 2107
Navy Annex
Washington, D.C. 20380

ATTENTION: Office of the Deputy Chief of Staff for Plans,
Policy and Operations (MD-P)
MAJ. T. L. Wilkerson (AA Room 2022) — 2

<u>DEPARTMENT OF THE AIR FORCE</u>

Office of the Deputy Chief of Staff, Operations,
Plans & Readiness
Department of the Air Force (XOX)
Washington, D.C. 20330

ATTENTION: Col. J. B. Donahoe, (Room 4C1061) — 2

<u>IDA</u>

Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

ADDRESSEE		NUMBER OF COPIES
IDA (continued)		
ATTENTION:	Alexander H. Flax	1
	Seymour J. Deitchman	1
	William J. Schultis	1
	Robert F. Robinson	1
	Lowell Bruce Anderson	1
	Stephen D. Biddle	1
	Rosemary H. Jones	1
	Edward P. Kerlin	1
	Dale L. Moody	1
	Leo A. Schmidt	1
	Eleanor L. Schwartz	1
	Harry Williams	1
	TIS	2
	ISAD Editor	1
	Mrs. E. Bailey	1
	TOTAL	248